(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

A NEURAL MODEL OF BILATERAL NEGOTIATION CONSISTING OF ONE AND TWO ISSUES

by

NEIL B. STRAND

SEPTEMBER, 1991

Thesis Advisor: Tung Bui

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL *(If applicable)* 367	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000	7b. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS *(City, State, and ZIP Code)*	10. SOURCE OF FUNDING NUMBERS Program Element No / Project No / Task No / Work Unit Accession Number

11. TITLE *(Include Security Classification)*
A Neural Model Of Bilateral Negotiation Consisting of One and Two Issues (UNCLASSIFIED)

12. PERSONAL AUTHOR(S) Strand, Neil B.

13a TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(year, month, day)*	15. PAGE COUNT
Master's Thesis	From To	1991, September	101

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES

FIELD	GROUP	SUBGROUP

18. SUBJECT TERMS *(continue on reverse if necessary and identify by block number)*
Neural network, Negotiation, Mental model of negotiation, Decision making

19. ABSTRACT *(continue on reverse if necessary and identify by block number)*

This thesis demonstrates that neural technology may be successfully employed to mimic some of the thought processes of a negotiator during a bilateral negotiation. Using the constraint satisfaction paradigm, originally developed to explore parallel distributed processing, a neural network is proposed to simulate the thought process of a buyer who negotiates the purchase of a good based on price and quality.

The findings of this thesis suggest that continued research in neural networks to replicate the mental model of the negotiator holds great promise. The ability to model true beliefs and evaluation methods has an advantage over more traditionaly prescriptive models. The neural network model allows incorporation of human irrationality and provides an ability to assess how that irrationality affects the negotiation outcome.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Tung Bui	22b TELEPHONE *(Include Area code)* (408) 646-2630	22c OFFICE SYMBOL AS/Bd

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

A Neural Model of Bilateral Negotiation Consisting of One and Two Issues

by

Neil B. Strand
Lieutenant, United States Navy
B.A., University of Minnesota, 1984

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
SEPTEMBER, 1991

Author: ______________________
Neil B. Strand

Approved by: ______________________
Tung Bui, Thesis Advisor

____
Lawrence Williamson, Second Reader

____
David R. Whipple, Chairman
Department of Administrative Sciences

ABSTRACT

This thesis demonstrates that neural technology may be successfully employed to mimic some of the thought processes of a negotiator during a bilateral negotiation. Using the constraint satisfaction paradigm, originally developed to explore parallel distributed processing, a neural network is proposed to simulate the thought process of a buyer who negotiates the purchase of a good based on price and quality.

The findings of this thesis suggest that continued research in neural networks to replicate the mental model of the negotiator holds great promise. The ability to model true beliefs and evaluation methods has an advantage over more traditionally prescriptive models. The neural network model allows incorporation of human irrationally and provides an ability to assess how that irrationality affects the negotiation outcome.

TABLE OF CONTENTS

List of Figures

I. INTRODUCTION

A. RESEARCH OBJECTIVE

This thesis explores the use of artificial neural network technology in a bilateral negotiation environment. Specifically, it seeks to model the negotiator's thought processes through the use of the constraint satisfaction paradigm.

B. RESEARCH BACKGROUND

Research in Negotiation Support Systems (NSS) has received much attention over the last few years. To support negotiation, a NSS often seeks to establish a consensual database as foundation for bargaining, help the involved parties evaluate the impact of their decision alternatives, search for agreements, and to provide communication links for facilitating discussion [Ref. 3:p. 689].

Based on the observation that the negotiation process is generally ill-structured, the majority of existing NSS focus on facilitating the communications between negotiators. Only a few of them attempt to model negotiation [Ref. 11:p. 142]. These attempts are primarily derived from economic models and game theory with the assumption that negotiators exhibit rational behavior. The objective of this thesis is to demonstrate that neural technology may be successfully employed to mimic some of the thought processes of a negotiator during a bilateral negotiation. We assume that the thought processes of the negotiator could be represented in a neural network and need not be rational.

C. RESEARCH METHOD

For this thesis, the artificial neural network is used to model thought processes in negotiation. The constraint satisfaction paradigm, developed by McClelland and Rumelhart to explore parallel distributed processing, is used as a vehicle to model the negotiator's thought process [Ref. 12:p. 38]. As discussed in Chapter III, the constraint satisfaction paradigm seeks to find a solution to a problem that requires simultaneous satisfaction to a very large number of constraints. These constraints are often interconnected and have different levels of importance.

D. SCOPE AND LIMITATIONS

Our focus is to develop a neural network that can model the thought processes of a negotiator. Due to the complexity of the neural network structure, this thesis considers only a bilateral setting with two agents. Furthermore, for the sake of clarity, it limits to two the number of negotiation issues. The proposed neural networks are implemented on the Parallel Distributed Processing software package copyrighted by McClelland and Rumelhart [Ref. 13:p. 356]. This thesis is not intended to be a comprehensive analysis of negotiation, nor is it a survey of the broad spectrum of neural paradigms rather, it seeks to use a neural network as a pattern matching scheme to represent the negotiator's mental model.

E. THESIS ORGANIZATION

Chapter II provides a brief discussion of issues of negotiation support systems. In Chapter III, a structure for building a neural network is presented. This structure will be used to develop the two bilateral negotiation neural networks discussed in Chapters IV and V. The first network will model the thought processes of a buyer negotiating with a seller over a single issue - i.e., price. The second neural network built in Chapter V will extend the single issue negotiation to include a second issue - i.e., quality.

II. NEGOTIATION, NEGOTIATION SUPPORT AND NEURAL NETWORKS

The purpose of this chapter is to introduce the reader to basic concepts in negotiation, negotiation support systems and neural networks.

A. ISSUES IN NEGOTIATION

Fisher and [illegible] have identified five factors which they believe impact the structure of negotiations. These factors are: [Ref. 7:p. [illegible]-98]

1. Separating the people from the problem
2. Providing communications between negotiators
3. Helping negotiators identify their real interests
4. Generating options for mutual gain
5. Use of objective criteria

1. The People and the Problem

The problem needs to be well addressed during a negotiation. However, diverse personal characteristics and psychological needs may disrupt a negotiation process by creating a focus on the person rather than the problem. Rules of negotiation and commitment to resolving differences can help create an environment where negotiation may progress to a solution. How an individual approaches conflict (contending, accommodating, compromising, collaborating or avoiding) may be overcome by creating an orderly, rational atmosphere which stresses equality and empathy with the opposing party [Ref. 8:p. 169].

2. Communications

Negotiators need to communicate. The success of a negotiation depends on the ability of negotiators to convey meaningful information to each other.

The way negotiators approach conflict influences the choice of negotiating environment. Accommodating and collaborative approaches tend to work better when both parties are in close proximity. Competitive types of negotiators may prefer to establish barriers which can exist as large distances between parties. [Ref. 8:p. 171]

3. Identify Real Interests

A negotiator needs to prepare for a negotiation by identifying and prioritizing goals and obtaining as much information on the opponent as possible. Comparison of interests with the opposing party may be accomplished through several methods. Multiple Criteria Decision Making, Game Theory, Conflict Analysis, Group Decision Theory, Generalized Approach for Structuring and Modeling Negotiations and Evolutionary Systems Design are methods of analysis which could be used by a negotiator. [Ref. 8:p. 174]

4. Mutual Gain Options

Some techniques for developing options are: brainstorming, interactive brainwriting pool technique, surveys and nominal group technique. Brainstorming is where small groups generate many solutions in a noncritical atmosphere. Interactive brainwriting avoids strong personalities in a group. The group members write their solutions rather than verbalizing them. Nominal group technique is somewhat of a mix between brainstorming and brainwriting. Each participant generates many written solutions which are later verbalized in small groups. [Ref. 8:p. 176]

5. Objective Criteria

Negotiation progresses from a point where negotiators have indistinct goals to a point where they have distinct, agreed upon goals or decisions [Ref. 8:p. 172]. Commonality of goals refers to how the negotiators arrive at their goals. How negotiators arrive at goals depends on their degree of cooperativeness [Ref. 8:p. 172]. They may act with a great deal of cooperation, some degree of cooperation or act non-cooperatively.

The accuracy and consistency of information used by negotiators can affect their decisions. Better decisions may be made by ensuring that accurate information is readily available [Ref. 8:p. 176].

B. Negotiation Support Systems

Negotiation support systems (NSS) are designed to assist negotiators in making rational decisions by providing a means of communication and through factual analysis of available information. To be effective a NSS should be customized to the individual so that the individual's needs can be taken into consideration in the analysis of alternatives and solutions.

Most of the NSS reported in the literature adopt a prescriptive approach in that they provide users with analytical models (e.g., multi-attribute utility theory, Nash solution, and Pareto optimization) to search for group solutions [Ref. 8:p. 178]. From that perspective, little effort is made to capture the true behavior of the negotiator that may contain irrational elements. Kersten and Szapiro attempt to provide a generalized approach to modeling negotiations. They assume that pressure constitute a primary driver in decision making. Such a modelling approach, although useful in understanding the causal effect of pressure on bargaining outcome, still suffers from providing a holistic description of the complex thought process of the negotiator. This thesis attempts to use the pattern matching concept in neural network to describe that thought process.

C. **Neural Networks**

Neural networks provide an unique means of seeking a solution to a problem. The problem and its solution may be represented as a pattern of activity among the elements of the neural network. Each element represents a priori information about the problem. When a set of elements are activated, the information associated with each individual element is combined in a manner that contributes to the formation of a solution. The elements are activated by matching a pattern of activity input to the network to a known pattern of activity among the elements which represent a priori information about that problem.

The neural networks presented in this thesis possess a priori information about how a negotiator will evaluate a given negotiation situation. Each neural element will represent an idea the negotiator has about what is important to making a decision. The collective activation of a set of these elements will simulate the process a negotiator would do in evaluating the negotiation situation according to what he/she perceived to be important.

The thought process of the negotiator - i.e., his/her preferences, perception of the problems, constraints and decision rules - exists in each neural network. Each network is capable of evaluating all the available options which the negotiator will consider in a manner consistent with his/her preferences. These networks do not provide analytical models of how a negotiator <u>should</u> make a decision. They provide models of how the negotiator <u>will</u> make a decision incorporating any irrationalities which the negotiator may possess.

III. BUILDING A NEURAL APPLICATION

A. STRUCTURE FOR BUILDING A NEURAL APPLICATION

Figure 1 shows the structure which will be used for neural network development. The first phase is to define the problem. The second phase consists of choosing a paradigm. During the third phase, the network will be constructed. The completed network will be tested in phase four.

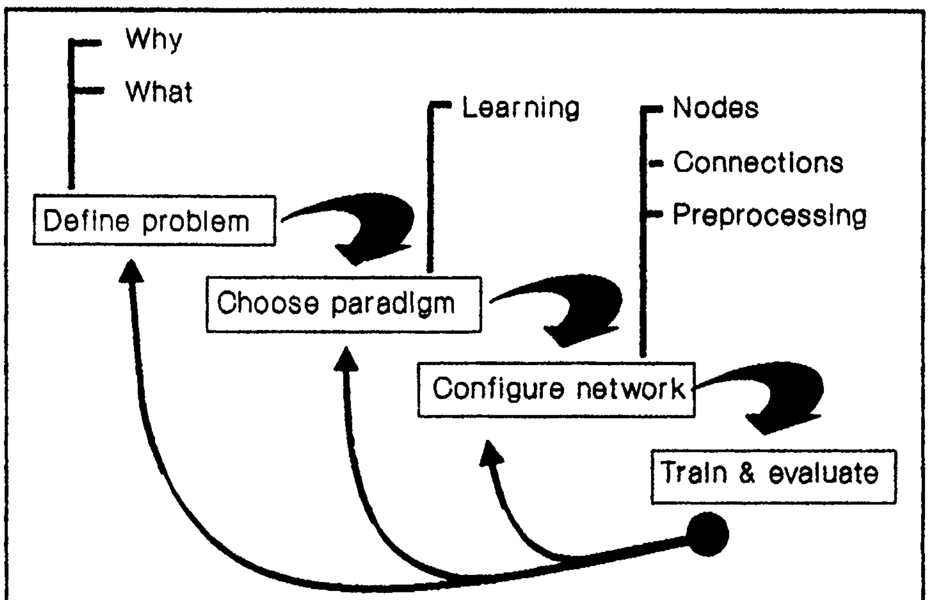

Figure 1 Neural network development model

B. DEFINE THE PROBLEM

When starting a neural network application development effort two questions should be addressed: 1) Why use a neural network? and 2) What will the network eventually do?

1. Why Use a Neural Network?

Often the need for a neural solution is based on the desire to experiment. It may be that a good solution exists but the idea of using a new approach intrigues the developer. Perhaps the present solution is difficult to execute and requires large amounts of computing power or several routines which are slow to develop the desired solution. Successful neural network applications have the following general characterizations: [Ref. 1:p. 38]

- Conventional computer technology is inadequate
- Problem requires qualitative or complex quantitative reasoning
- Data is readily available but multivariate, noisy or error prone
- Solution is derived from highly interdependent parameters which have no precise quantification
- Algorithmic solution is unknown, difficult or expensive

So why choose a neural net? There are some problems such as those characterized above which are well suited for a neural solution. Comparison with an expert solution or an algorithmic solution must be weighed. Neural networks work very well with identifying patterns. They degrade "gracefully" when compared with other solutions [Ref. 14:p. 472]. In an associative memory application employing distributed representations, the neural net can withstand a loss of part of itself or of the input pattern and remain functional. This is due in part to the natural redundancy which a distributed memory possesses [Ref. 14:p. 472].

This degradation effect reflects a key concept of neural networks. A neural net "stores" it's knowledge in it's pattern of connections. Recall of a piece of information is done reconstructively, through activation of the appropriate nodes [Ref. 9:p. 36]. A neural net can learn which nodes should be activated for a particular pattern. When that pattern or one similar to it is presented to the system, the associated level of activity is generated throughout the system resulting in a best guess response. The memory of the system exists in the whole system, not within a specific node and exists only during a certain level of excitement within the system.

2. What the Neural Network will do?

Once it is decided that the neural network approach may be appropriate for our application, the problem must be developed. Identifying what the neural net needs to do and identifying input and output, will simplify choosing the appropriate paradigm and creating the appropriate environment for the network to work in.

This task is best approached from a top down perspective, where a general problem description will be developed followed by a detailed analysis. The detailed analysis should describe the desired properties and learning methods of a neural net.

Suppose an optical character recognition application (OCR) is to be built. The application would need to recognize pixel patterns and convert them to ASCII format. This general concept may be broken down into elements. Pixel patterns would be associated with ASCII elements. The pixel patterns could be classified according to their features and these features would elicit the appropriate ASCII response. Therefore a feature detector and a classifier type of network would be necessary. Since the input and output association can be controlled, supervised learning can be done.

It would be possible to employ two networks, one to detect features and another to classify these features but a single network is desired for simplicity.

When it is decided what the network will do, the input and output needs to be considered. During this process, the problem may be classified. According to Caudill, there are 5 general types of problems: mapping, Associative memory, categorization, temporal mapping and image processing [Ref. 4:p. 30].

A mapping problem associates an input pattern with an output pattern [Ref. 4:p. 30]. This is not reproducing the exact pattern but includes the ability to generalize to something close to what the network was trained for. This type of problem closely resembles that which the pattern associator paradigm solves, described in McClelland and Rumelhart [Ref. 14:p. 161]. Caudill recommends backpropagation and counterpropgation networks to work with this type of problem [Ref. 4:p. 32].

An associative memory problem is reproduction of a pattern [Ref. 4:p. 32]]. Basically this is stored information which is recalled upon input of the associated information. The auto associator paradigm would be a subset of the paradigms which work with this type of problem [Ref. 14:p. 161]. Caudill includes backpropagation, counterpropagation and Kohonen types of networks for this problem [Ref. 4:p. 32].

Categorization problems, classify input patterns into categories [Ref. 4:p. 32]. This would allow several different inputs to cause the network to respond in the same manner, as long as the inputs were of the same category. Both the classification and regularity detector paradigms could be included in solving this type of problem [Ref. 14:p. 161]. According to Caudill, Kohonen and adaptive resonance networks are good choices for this problem [Ref. 4:p. 32].

Temporal mapping problems include the element of time in the input pattern [Ref. 4:p. 32]. This is often the case with process control applications. Backpropagation or any recurrent network are recommended by Caudill [Ref. 4:p. 32].

Image processing problems are in a separate class according to Caudill due to the significant amount of data which must be input, otherwise they are similar to a mapping problem [Ref. 4:p. 32].

C. CHOOSE THE PARADIGM

Choosing a paradigm is largely a matter of eliminating those paradigms which are clearly unsuitable for the problem [Ref. 4:p. 30]. The backpropagation paradigm is a choice for nearly all problems except the categorization problem. The availability of training data though, may make the backpropagation network unsuitable. For a backpropagation network to learn, it needs to be supervised during training.

Three common methods of learning are supervised, unsupervised and reinforced. [Ref. 1:p. 41]. Supervised learning is a method where a pattern is presented to the system along with the desired result or teaching pattern [Ref. 1:p. 41]. This method of learning results in a system which learns to associate patterns. Unsupervised learning is used to develop a regularity detector [Ref. 14:p. 57]. Patterns are presented to the system without a teaching pattern. The system is allowed to develop it's own representation of the input based on the features which it determines are appropriate. The reinforcement method of learning does not explicitly provide a correct teaching pattern. Instead, the system is directed to the desired result by reinforcing good outputs similar to a grading scheme [Ref. 1:p. 41]. This method is relatively uncommon due to it's complexity.

D. CONFIGURE THE NETWORK

After the paradigm is selected, designing of the network may begin. During this phase, the problem is reviewed and developed into a network which possesses the learning properties of the selected paradigm.

1. General Configuration/Update Mechanisms

The components of a network describe the properties which a network possesses. They describe how a network learns, how signals are distributed throughout the network, and how a network will react to a signal after it learns.

According to Rumelhart, Hinton and McClelland, there are eight major components of a parallel distributed processing model. They are: [Ref. 14:p. 46]

1. A set of processing units (PU's)
2. A state of activation (a_i)
3. An output function for each unit
4. A pattern of connectivity among units (w_{ij})
5. A propagation rule
6. An activation rule
7. A learning rule
8. An environment

A set of processing units form half of the physical representation of a neural net. A PU may be thought of as a simple processor which receives input and calculates an output value. The PUs can be classified into three categories which describe their general source of input and the destination of their output. There are input PUs which generally receive their input from an external source and direct their output to hidden or output PUs. Hidden PUs receive their input from other PUs and direct their output to output PUs. Output PUs receive their input from other PUs and direct their output outside of the system. [Ref. 14:p. 48]

Hidden PU's are used to develop internal representations of the data for the network. In some problems, the system cannot learn without hidden PU's.

The state of activation is a time dependent measure of the pattern of activity of a system. Each unit (i) has an activation state at time t designated as $a_i(t)$. The value of this activation function may be discrete or continuous. [Ref. 14:p. 48]

The output of a unit is determined by its degree of activation. It is used to communicate with a unit's neighbors. The output function may be an identity function in which case the output is equal to the activation $o_i(t) = F_i(a_i(t))$ or it may be a threshold function where a unit does not interact with its neighbors unless it is activated by a certain amount. The output function may also be stochastic in nature where its activation affects the probability of its output. [Ref. 14:p. 49]

The pattern of connectivity is the other half of the physical make up of a system. It describes how the PU's are connected to each other. It is this pattern which embodies the knowledge of the system and dictates how the system will react to stimuli. The pattern of connectivity is specified by assigning weights (W_{ij}) to each connection between PUs. A positive weight indicates a reinforcing or excitatory connection while a negative weight represents a contrary or inhibitory connection between nodes. The connections provide the means for calculating the total sum of the input to a unit. This sum is calculated as the weighted sum of each neighbor's activation level where the connection strength provides the weighting factor. [Ref. 14:p. 49]

The pattern of connectivity forms the basis of the knowledge of the network. The complexity of the pattern ranges from simple additive contributions to complex sets of connection matrices. [Ref. 14:p. 49]

The rule of propagation specifies how the output vector O(t) from a set of units will be combined with the connectivity matrix W, to produce the net input for each type of input to a unit. For simple systems this rule can be simply stated as the weighted sum of the inputs to a unit, $net_e = W_eO(t)$ where net_e is the set of excitatory inputs to a unit and W_e is the excitatory matrix. [Ref. 14:p. 51]

The activation rule determines the method with which the current state of a unit and the net inputs (net_i) to that unit will produce a new state of activation. In the simple case there is an identity function F where the activation of a unit at time t+1 is equal to the net input to that unit at time t, $a_i(t+1) = F(net_i(t))$ where $F = 1$. Often the previous activation level, $a_i(t)$, of a unit will be included in the function. It may be that the function is a threshold function where $a_i(t)=1$ if the total input net_i exceeds some value. There are other cases where F is stochastic or a decaying function. [Ref. 14:p. 51]

Often the activation rule and the output rule are combined into what is called a transfer or squashing function [Ref. 1:p. 43].

Learning can consist of three modifications of the connections: developing new connections; removing existing connections; modifying the strength of connections [Ref. 14:p. 52].

Most learning rules of this type are considered variations of the Hebbian learning rule proposed in 1949. Hebb's basic concept is that if a unit receives input from another and if both are highly active, then the weight between both units should be strengthened. In the simple case of learning the learning rule is the function: [Ref. 14:p. 53]

$$\Delta w_{ij} = \eta a_i o_j$$

Where η is the constant of proportionality representing the learning rate, o_j is the output vector from unit j to unit i and a_i is the state of activation of unit i [Ref. 14:p. 53].

The environment which is typical in the parallel distributed processing model is characterized by a stable probability distribution over the set of possible input patterns independent of previous inputs and responses of the system. In other words, there is some probability that any one of the possible set of inputs is affecting the input nodes at a particular time. [Ref. 14:p. 54]

When designing the network, each of these characteristics must be accounted for. How many PUs will be used, the pattern of connectivity, the transfer function, learning ability and the environment can have significant effects on the success of the implementation.

2. Nodes

Determining the number of nodes in a neural net depends on the type of representation used. Distributed representations allow each node to represent more that one entity [Ref. 14:p. 77]. The choice of using distributed representations offers advantages such as reducing the node count and ability to sustain damage to a node without losing a large amount of information [Ref. 14:p. 472].

A local representation scheme would necessitate that there be a unique node for each possible value [Ref. 14:p.

94]. A plane with ten discrete values possible in each direction would need 100 feature nodes to be fully represented. By dividing the plane into several overlapping zones, each represented by a node, the number of nodes required to represent the plane is reduced [Ref. 14:p. 91]. Obviously, there is a resolution/accuracy tradeoff which must be considered [Ref. 14:p. 93].

If there are several closely spaced points of interest on this plane, the zones may not be able to discriminate between individual points. By reducing the feature space (i.e., decrease the density of the points) the overlapping zones would be able to discriminate to a greater degree [Ref. 14:p. 92]. This concept reflects the need to accurately assess the boundaries of the problem and data.

3. Connections

The pattern of connectivity of a system represents the knowledge contained within that system. The pattern of connectivity denotes which units are connected to each other and the strength of that connection. The strength of the connection from unit j to unit i and is represented by a weight, W_{ij}. If a unit reinforces another the weight will be positive while if the units contradict each other, the weight will be negative. [Ref. 14:p. 49]

Defining the correct connections within a network is critical to fully understanding what the network will be able to do. As the previous paragraph indicated, the knowledge of the system exists in the connections. If the system is to act as a memory device, the nodes will not contain specific recall values, but the interactions between the nodes will create the recall value [Ref. 4:p. 3]. The desired value does not exist except in a surreal sense, present only when the appropriate level of activity in the system is achieved.

4. Preprocessing

While not strictly a part of the network, preprocessing of the input data can be critical to achieving success. Preprocessing consists of normalizing, parameterizing, scaling or otherwise manipulating the input data into a workable form.

If there are several channels of data feeding into the system, care should be taken with normalizing the data [Ref. 6:p. 70]. For instance, if there are two channels which have a normal range of 0.02 and two channels which have a normal range of 20.0, it would not be prudent to normalize across all channels using the same base. In this case, normalizing the two low range channels independent of the high range channels would allow each channel to provide a reasonable input. Normalizing across all channels would preclude the low range channels of having a significant influence on the network.

If there are many input channels, there may be a method of combining or categorizing several which have similar parameters or meaning. Care must be taken to preserve the data features present in each separate channel.

E. TRAINING AND TESTING

Training the network is largely a matter of presenting the network with the preprocessed data (and teaching data if supervised) until it learns to an acceptable level of accuracy. The method of presenting the data to the network may affect how well it learns. The data may be presented sequentially or randomly. In packaged networks, there are usually a set of parameters which allow adjustment of the speed at which a network learns.

Evaluating the trained network may be done several ways. Testing data may be input, and the results then analyzed. The weights of the connections may be analyzed. If the connection weight is zero, for all connections to a particular node, that node is probably insignificant and it's removal should be considered. Individually activating input nodes, and then tracing the pattern of activity may reveal unexpected or undesired relations existing in the network. The meaning of each node should be evaluated in this manner. For example, if a node representing a bad credit rating activates the node representing credit approval, something is wrong. [Ref. 2:p. 38]

If testing shows an error in the network, the first thing that should be evaluated is the training [Ref. 2:p. 38]. If the training appears to be sound, then the design of the network should be analyzed. Finally, the network may be inappropriate for the problem and a different paradigm might be the choice, the problem may have a structural flaw or the problem is not suited to a neural application.

IV. BUILDING A NEURAL MODEL FOR NEGOTIATION BASED ON A SINGLE ISSUE

This chapter will describe the process of building a neural network for negotiation based on a single issue. This process will follow the structure described in chapter III.

A. DEFINE THE PROBLEM

Why is a neural solution sought?

A neural solution to a single issue negotiation problem will act as a stepping stone in the development of a more comprehensive neural solution to negotiation.

What will the neural network have to do?

The network will need to simulate the logical decision making processes of a negotiator. The ability to simulate these processes will depend on capturing physical representations of the environment within the network as schema. The network will be limited to the buyer's perspective.

This model will contain four general representations. They are:

1. The buyer's current offer
2. The seller's current offer

3. A comparison of the offers

4. The buyer's response to the seller's offer

These representations will serve as bits of knowledge or parts of the state of the environment, which exist in the buyer's mind during negotiation. A set of schema will exist when each of these representations are present in the network. To encode these representations in a neural network it is helpful for the developer to imagine each representation as part of a "picture" which exists within the mind. The picture is a schema and each representation is part of that schema.

In a hypothetical process, the buyer will "see" his offer as a value. Likewise, he will also "see" the seller's offer as a value. Both of these values contribute to the picture which is forming in the buyer's mind. The buyer will compare his own offer with that of the seller. The idea of comparison exists in the buyer's mind before the offers are made. This comparison idea will only have real meaning when the buyer is able to "see" both offers. When both offers exist, a comparison will exist. The buyer's mind will fill in the idea of comparison to form a larger part of the picture. To complete the picture, the buyer's mind will select a response offer. The selection of a response offer will complete the picture <u>and</u> convey an action on the buyer's environment.

To achieve the above action, a neural network will have to act like a memory device containing many small parts which can be connected together to create pictures of the possible states of the environment. Whenever the environment changes (i.e., new seller's offer), a new picture of the environment will be formed. A single picture will contain both negotiators' offers, a view of the comparison of the offers and a response offer.

Creating a simulated memory which would fill in the necessary missing pieces can be done by establishing relations between each part so that a set of parts may become a whole picture. Each part can be considered as a hypothesis of the environment. Each hypothesis may contribute, detract or have no effect on the validity of the whole picture. The relations between the hypotheses can be considered as constraints. Hypothesis A and hypothesis B could have a constraint between them which implies that either both or neither must exist at the same time. Hypothesis C and hypothesis D may have the constraint that only one of them may exist at the same time.

B. CHOOSE THE PARADIGM

The engine chosen for this model is the Constraint Satisfaction (CS) network provided in "Explorations in Parallel Distributed Processing" [Ref. 13]. The nodes, connections and screen layout are auxiliary files developed with a text editor. This network does not possess any learning capability [Ref. 13:p. 54]. The nodes and connections must be established manually. This feature is good since control is maintained over the set of schema which will be represented in the network. A learning mechanism might change the structure of the schema, producing unwanted results.

CS networks work on the principle that each node is a hypothesis and connections between nodes are constraints between hypotheses [Ref. 13:p. 50]. The CS network which Rumelhart and McClelland developed is designed to work with weak constraints.

> "PDP constraint networks are designed to deal with *weak constraints* (BLAKE, 1983), that is, with situations in which constraints constitute a set of desiderata that ought to be satisfied rather than a set of hard constraints that must be satisfied." [Ref. 13:p. 50]

The importance of the constraint may be coded into the weight of the connection. An important constraint would have a large value while those not so important would have lessor values. [Ref. 13:p. 50]

Rumelhart and McClelland view external input as evidence to a hypothesis [Ref. 13:p. 50]. The CS network has a feature which "clamps" a node on if the input is positive or off if the input is negative [Ref. 13:p. 57]. This allows for some processing of hard constraints where the hard constraint is the external input to a node.

If there is prior evidence that a hypothesis may be true or false, that evidence can be represented by assigning a bias term ($bias_i$) to node i. The bias will activate (positive) or deactivate (negative) a node in the absence of other evidence. [Ref. 13:p. 50]

The connection weight, w_{ij}, is the strength of the connection to node i from node j [Ref. 13:p. 6]. In the CS network the connections are symmetric, (i.e., $w_{ij} = w_{ji}$) and a node may not connect with itself (ie. $w_{ii} = 0$). [Ref. 13:p. 53]

The state of activation of node i at time t is $a_i(t)$. The state of activation is updated according to the following equations: [Ref. 13:p. 53,54]

$$a_i(t+1) = a_i(t) + net_i(1-a_i(t))$$
$$\text{if } net_i > 0$$
$$\text{and}$$
$$a_i(t+1) = a_i(t) + net_i(a_i(t))$$
$$\text{if } net_i < 0$$

The net input to a node (net_i) is determined according to the following equation [Ref. 13:p. 54].

$$net_i = istr(\sum_j w_{ij}a_j + bias_i) + estr(input_i)$$

The istr term is a scaling term which affects the network generated inputs to a node. The estr term is a scaling factor which affects the external input to a node. [Ref. 13:p. 257,258]

Nodes are randomly updated asynchronously within the network [Ref. 13:p. 55]. This method of update was developed by Hopfield [Ref. 14:p. 61]. Random, asynchronous update has the advantage of maintaining better stability by reducing oscillations between states, than synchronous update where all units are updated at the same instant [Ref. 14:p. 61].

Rumelhart and McClelland use the term "goodness of fit" to describe how well a CS network satisfies constraints [Ref. 13:p. 50]. This measure depends on three factors.

> "First, it depends on the extent to which each unit satisfies the constraints imposed on it by other units. ...Second, the a priori strength of the hypothesis is captured by adding the bias to the goodness measure. Finally, the goodness of fit for a hypothesis when direct evidence is available is given by the product of the input value times the activation value of the unit." [Ref. 13:p. 50,51]

Therefore, equation below is used to measure the goodness of a single node within a network [Ref. 13:p. 51].

$$goodness_i = \sum_j w_{ij} a_i a_j + input_i a_i + bias_i a_i$$

The overall goodness of the network can be measured with [Ref. 13:p. 51].

$$goodness = \sum_{ij} w_{ij} a_i a_j + \sum_i input_i a_i + \sum_i bias_i a_i$$

During processing, the network will maximize its goodness of fit measure. External inputs will provide a stimulus to a set of nodes. The remaining nodes will or will not be activated depending on how they affect the overall goodness measure.

C. CONFIGURE THE NETWORK

This network will simulate the decision making processes of a buyer attempting to purchase a bicycle. The model assumes that a "soft" negotiation environment exists. A soft negotiation is one where both parties attempt to cooperate to find a mutually beneficial agreement [Ref. 8:p. 168].

A single issue will be negotiated. The issue has six discrete alternatives all of which are acceptable to the buyer. The buyer will attempt to purchase the bicycle for the lowest amount possible but he will readily concede some amount to maintain a good relationship with the seller.

The buyer has three objectives. 1) Reach an agreement by conceding $10. 2) Move closer to seller's position quickly. 3) Induce seller to reduce price. These objectives will be used by the buyer during the negotiation to evaluate alternatives [Ref. 10:p. 4].

The buyer model has 5 strategies built into it. They are:

1. If there is no current buyer offer, offer $100.

2. If both buyer and seller offer same amount, accept that amount.

3. If the buyer and seller differ by $10, accept the seller's price.

4. If the buyer and seller differ by more than $10, the buyer will concede $20 only if he is currently offering $100 or $110.

5. Once the buyer offers $120, only $10 will be given up at a time until agreement is reached.

Strategy 1 and 2 are default strategies and are self explanatory. Strategy 3 reflects the first objective of the buyer. The buyer does not perceive much difference in his aspiration levels between two outcomes when an agreement may be reached by choosing the outcome corresponding to a lower aspiration level.

Strategy 4 reflects the second objective of the buyer with the caveat that the buyer does not want to concede too much to the seller. Essentially the buyer is offering to "split the difference" with this strategy. The aspiration levels between the highest and that of the $120 outcome do not differ much. The buyer does not change the value he has for owning the bicycle very much.

Strategy 5 reflects the third objective. Once the buyer has conceded $120 or more, he expects the seller to be cooperative. At this point, the buyer's aspiration levels decline rapidly since his perceived value of owning the bicycle declines rapidly once the price becomes more than $120. At this point, the buyer realizes that the seller possesses more power in the negotiation than the buyer. The buyer will seek justice by only conceding $10. This may send a message to the seller that the price of the bicycle is getting expensive.

Each of these strategies is an action on the environment as the buyer perceives it. Therefore, the model must be capable of perceiving the environment in terms of the buyer's current offer, the seller's current offer and a comparison of the offers.

Five groups of nodes or representations are needed to create the representations needed for the neural network to perceive the environment. Four of these representations have been presented in the problem definition phase as representations. A fifth representation is needed to in order to represent the idea that the buyer may not have made an offer. These groups of nodes are:

1. The buyer's current offer
2. The seller's current offer
3. A comparison of the offers
4. The buyer's response to the seller's offer
5. The absence of a buyer offer

Each of the representations will be broken into a set of nodes (hypotheses) and connections (constraints) corresponding to the discrete values possible during the negotiation. Each node will have a continuous activation range from zero to one.

We assume that the buyer would prefer to pay only $100 but will pay $150 if necessary. The buyer knows that the seller will ask a maximum of $150 and may consider $100 acceptable. Therefore, the buyer's current offer and the seller's current offer may be broken into twelve individual nodes, six per person (indicated by black dots shown in Figure 2). At this point the only connections are inputs from the external environment.

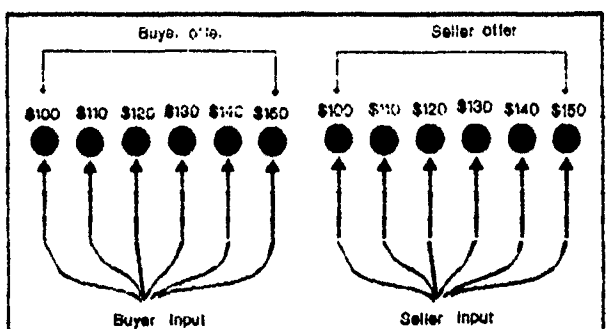

Figure 2 Buyer and Seller input representations

The buyer will only be concerned with the difference between his current offer and the seller's current offer for making a decision as to what next to offer.

To represent the difference in offers, it is necessary to have twenty one representations. Six representations are necessary for a buyer offer of $100. They are (buyer, seller); ($100,$100), ($100,$110), ($100,$120), ($100,$130), ($100,$140), ($100,$150). Five representations are necessary for a \$110 buyer offer; (\$110,\$110), (\$110,\$120), (\$110,\$130), (\$110,\$140), (\$110,\$150). Following the same line of thought, four representations are necessary for the buyer \$120 offer, three for the \$130 offer, two for the \$140 offer and one for the \$150 offer. Figure 3 shows these comparisons. Each combination of buyer/seller/comparison node is a single representation.

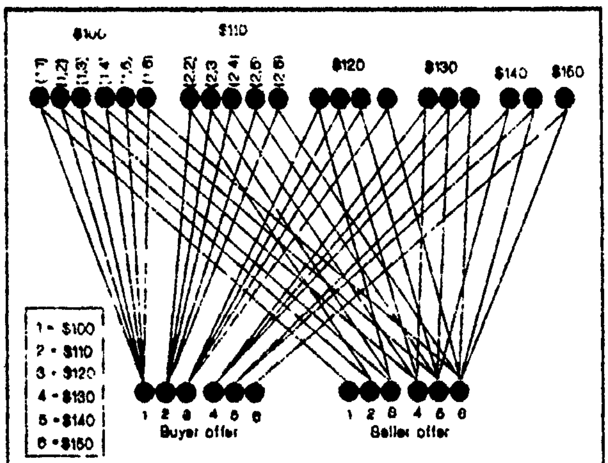

Figure 3 Individual comparison representations

The buyer will not be concerned with the exact difference in offers when a seller's offer is \$20 greater than the buyer's offer. The number of comparison nodes therefore, may be reduced to fifteen. Three comparison nodes will be necessary for each buyer offer less than \$140 (agree, disagree by \$10, disagree by \$20 or more). Two comparison nodes will be necessary for the \$140 buyer offer (agree, disagree by \$10). The \$150 offer will require one comparison node (agreement).

Figure 4 shows how the number of comparison nodes would be reduced. "A" represents agreement, "D" represents disagreement by $10, "D+" represents disagreement by two or more dollars. There are still twenty one representations or "pictures". The D+ nodes act as concentration points for a number of representations. For example, the $100 D+ node completes four possible representations ($100,$120), ($100,$130), ($100,$140) and ($100,$150). Conceptually, the buyer has categorized the environment into fifteen different views or pictures.

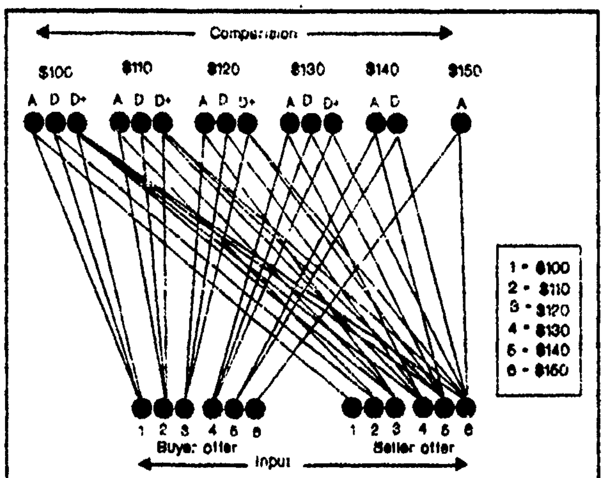

Figure 4 Combined comparison representations

After the initial offers are established, the buyer will increase an offer based on the difference between his and the seller's offer. If there is no difference, the buyer will accept that offer (strategy 2). If the seller asks a $10 more than the buyer's current offer, the buyer will meet the seller's price (strategy 3). If the buyer is offering $100 or $110 and the seller is asking $20 or more than the buyer is offering, the buyer will increase the offer by $20 (strategy 4).

After the buyer has reached the $120 point, only a $10 will be conceded (strategy 5). Figure 5 shows the relations between the comparison nodes and the desired responses. There are fifteen different views of the environment but only six different actions which the buyer may take on the environment.

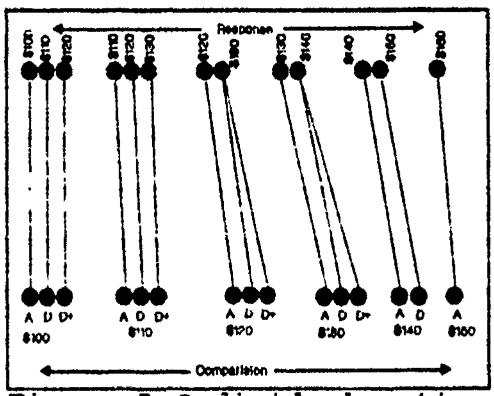

Figure 5 Individual action responses

Since there are only six possible actions or response values ($100 - $150), the responses may be merged together as shown in Figure 6.

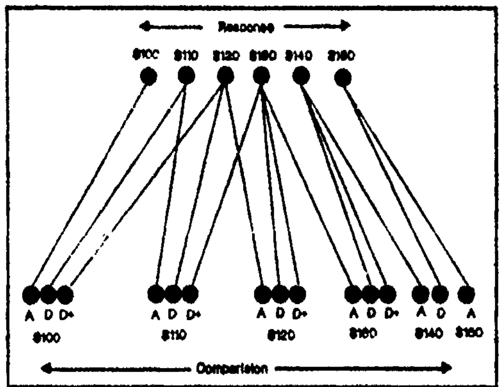

Figure 6 Combined action responses

A special case is when the buyer may not have made an offer, preferring to let the seller begin negotiation. A representation of the absence of a buyer offer is necessary. The no offer node in combination with all of the buyer input nodes will create the representation of the absence of a buyer offer. The desired action in the absence of a current buyer offer is for the buyer to offer $100 (strategy 1). If any of the buyer nodes are active, the no offer node must be inactive. Figure 7 shows how this idea is represented.

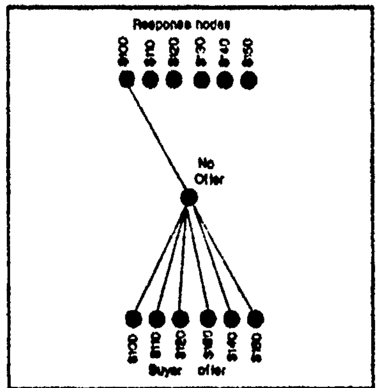

Figure 7 No offer representations

Since both a buyer and a seller input are necessary to activate an appropriate comparison node, the connection weight should be set so that activation of a single input node does not activate a comparison node. All weights are symmetrical in a CS network. If a single input node were allowed to activate a comparison node a comparison node would be able to activate an input node, the entire set of nodes would soon become active. There is no automatic decay mechanism in the CS paradigm. When a node becomes active, it will stay active unless it is specifically instructed to do otherwise.

There are two tools available to control activation. Inhibition connections and negative bias. Inhibition connections could be established between each of the nodes in each set (ie. each buyer node inhibits all of the other buyer nodes, each comparison node inhibits all other comparison nodes, etc). A combination of negative bias and connection weight could also be used. Negative bias ensures that a node remains inactive until the weighted activation (Σwa) becomes greater than the negative bias.

For this model negative bias will be used. It is simpler to modify than inhibition connections. Each of the input and comparison nodes will be assigned a bias of -0.7. The connection weight between the inputs and the comparison nodes will be +0.5. Between the comparison and response nodes the connection weight will be +0.1. No bias is necessary for the response nodes. The bias value of -0.7 ensures that an input node will not be erroneously activated by a single active comparison node which is connected to it and that a comparison node will not be activated by a single active input node. The discussion below addresses this problem.

The equation below shows how the activation of the \$100 D+ node will be computed (see constraint satisfaction description). The buyer and seller are limited to a single offer at a time. When that offer is made, the activation level of the input nodes will be "clamped" to 1.0.

$$\sum_0^{12} 0.5a_{input} = 0.5a_{B1} + 0.5a_{S3} + 0.5a_{S4} + 0.5a_{S5} + 0.5a_{S6} + 0.1a_{r3}$$

$$net_{D+} = \sum_0^{12} 0.5a_{input} + (-0.7)$$

$$\textit{when } net_{D+} > 0 \textit{ then}$$

$$a_{D+}(t+1) = a_{D+}(t) + net_{D+}(1 - a_{D+}(t)$$

$$\textit{else when } net_{D+} < 0$$

$$a_{D+}(t+1) = a_{D+}(t) + net_{D+}a_{d+}(t)$$

In this example, assume that the current offers are (\$100,\$120) and time t=1 for the comparison node. The *\$100 D+* node will receive a net input of 0.3 (1.0 + -0.7). No other node will receive a net input greater than 0 due to the negative bias associated with it. The activation level of this *D+* node (assuming that it was previously inactive) will become 0.3 (0 + 0.3(1 - 0)).

When this *D+* node becomes fully active (1.0) it will not be able to activate any input node. The maximum net input to any other input node (except 1,3) is -0.2 (0.5 + -0.7).

When the *D+* node becomes active, it will send an activation signal to the *\$120 response* node. The symmetrical connections then allow the *\$120 response* node to reinforce the *\$100 D+* node. The same symmetrical connections also allow the *\$120 response* node to send an activation signal to the *\$110 D* comparison node. The *\$110 D* comparison node will receive activation signals from the seller *\$120 offer* node and the *\$120 response* node. This comparison node will receive a net input of -0.1 (0.5 + 0.1 + -0.7).

The *no offer* node is biased slightly on (+0.1). The connection weight between it and the buyer input nodes is strongly inhibitory (-1.0). The connection weight between it and the *\$100 response* node is positive (+0.1).

The completed model consists of a parallel architecture consisting of 34 nodes and 53 symmetric connections. Figure 8 shows the layout of the connections and nodes. For clarity, some of the connections from the *buyer* and *seller* input nodes to the feature nodes are not shown.

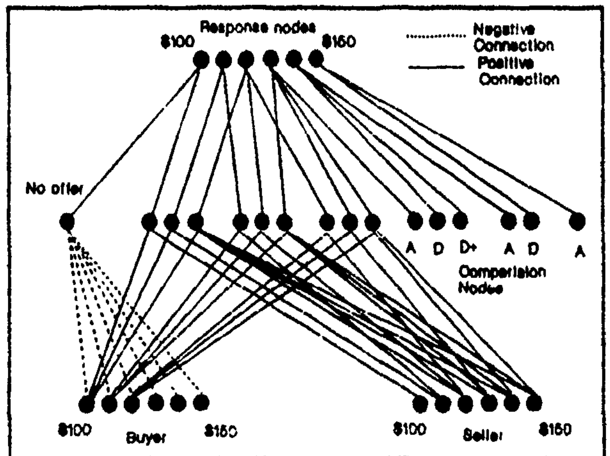

Figure 8 Network Architecture

The *buyer* and *seller* nodes act as binary nodes due to clamping. Each node represents a specific number of dollars each participant is currently offering. The *response* nodes respond with the new dollar amount which the buyer should offer to the seller. Each node corresponds to a specific dollar amount. The *response* nodes react to the activation of specific *comparison nodes*.

Figure 9 shows the single issue network file which is used by the CS program to configure the network. Each row represents the weights to that node of the network [Ref. 13:p. 266]. For example, the a in the first row represent the weight of the connection from the *$100 A* node to the *$100 buyer input* node.

The first six rows correspond to the *buyer input* nodes. The next six rows are the *seller input* nodes. Row 13 through 27 represent the *comparison* nodes. Row 28 through 33 are the response nodes. The last row represents the no offer node.

All weights are symmetric. The **a** in row 1 corresponds to the first **a** in row 13. The **b** in row 1 corresponds to the first **b** in row 14. Each letter is a symbol representing the weight assigned to that symbol in the constraints section immediately preceding the network description portion.

The use of a,b,c is for simplicity in debugging, each of these symbols has the same weight value. They correspond to the connections between the *input* and *comparison* nodes. The **d** symbol corresponds with the weight between the *comparison* nodes and the *response* nodes. The **x** symbol corresponds to the negative bias of the *input* and *comparison* nodes. The **e** is the weight between the *buyer input* and the *no response* node. The **g** is the weight between the *no response* and the *$100 response* node. The **h** is the bias of the *no response* node.

```
definitions:
nunits 34
end
constraints:
a 0.5
b 0.5
c 0.5
d 0.1
x -0.7
e -1.0
g 0.1
h 0.1
end
network:
............abc..................e
...............abc...............e
..................abc............e
.....................abc.........e
........................ab.......e
..........................a......e
............a.....................
.............b.a..................
..............c.b.a...............
..............c..c.b.a............
..............c..c..c.b.a.........
..............c..c..c..c.ba.......
a.....a....................d......
b......b....................d.....
c.......cccc.................d....
.a.....a....................d.....
.b......b....................d....
.c.......ccc..................d...
..a.....a....................d....
..b......b....................d...
..c.......cc..................d...
...a.....a....................d...
...b......b....................d..
...c.......c...................d..
....a.....a....................d..
....b......b....................d.
.....a.....a....................d.
............d....................g
.............d.d..................
..............d.d.d...............
.................d.ddd............
......................ddd.........
.........................dd.......
eeeeee.....................g......
end
biases:
xxxxxxxxxxxxxxxxxxxxxxxxxxxx.....h
end
```

Figure 9 Network Description File

D. TEST THE NETWORK

To develop an offer if there is no current offer, the *no offer* node is biased (+0.1) to be active. It has a small weight (+0.1) connecting it to the first *response* node. The connection with the *buyer* nodes is negative (-1.0) which will effectively turn the node off if the buyer has a current offer.

If there is no current offer by the buyer, The *buyer* nodes will be inactive. Figure 10 shows the connections and nodes involved with this state of activity. The empty nodes represent inactivity while the filled in nodes are active.

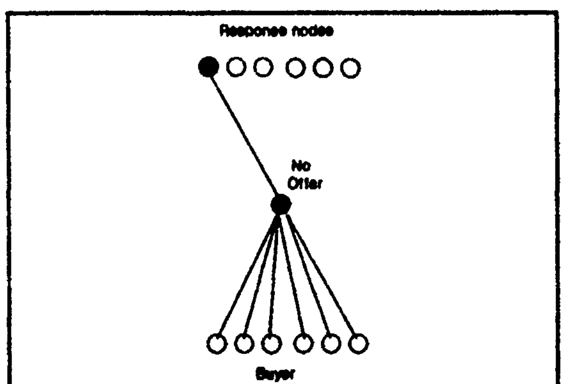

Figure 10 No Offer

The output screen is shown in Figure 11. Only the *$100* response offer on the far right is active. The level of activity of the no offer node is not displayed.

```
disp/ exam/ get/ save/ set/ clear cycle do input log
newstart quit reset run test

 (buyer model)                 disagree   disagree   recommended
bid      price      agree       weakly     strongly     bid

0  b     0 one      0 a1       0 d1       0 d1+      9 one
0  u     0 two      0 a2       0 d2       0 d2+      0 two
0  y     0 three    0 a3       0 d3       0 d3+      0 three
0  e     0 four     0 a4       0 d4       0 d4+      0 four
0  r     0 five     0 a5       0 d5                  0 five
0        0 six      0 a6                             0 six

0  s     0 one
0  e     0 two
0  l     0 three
0  l     0 four
0  e     0 five
0  r     0 six
```

Figure 11 Buyer offers one dollar

When the buyer and seller differ by $20 or more, the network will attempt to seek a quick compromise only if the buyer is offering $100 or $110. The *100D+* & *110D+* nodes and connections represent this idea. These nodes are biased (-0.7) to be inactive unless both the *buyer* and *seller* nodes which are connected with them are active. The connection strength between the *D+* nodes and the *buyer* and *seller* nodes is +0.5 and the connection strength with the *response* node is +0.1. The connection strengths are the same for all input/feature and feature/response connections. Figure 12 shows the set of connections which develop this idea.

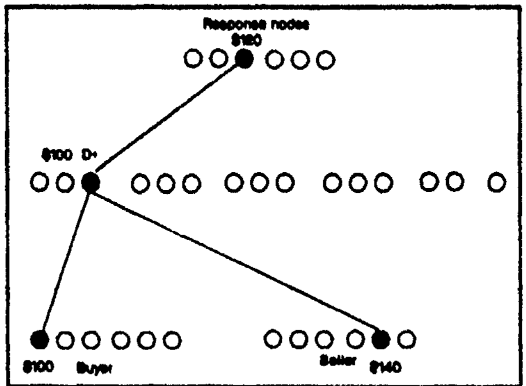

Figure 12 Seek quick agreement

In Figure 12, the active nodes are filled in. The *100D+* node detects that the buyer is offering $100 and the seller is asking $120 or more. This node and all of the active nodes connected to it, represent the schema in which the buyer is offering $100 and the seller is asking $120 or more and, the buyer should increase his bid by $20. When the *$120 response* node becomes active a complete schema is formed. The network has effectively filled in the missing piece of the schema by activating the *$120 response* node.

The output screen for this state of activation is shown in Figure 13. The active nodes are underlined.

```
disp/ exam/ get/ save/ set/ clear cycle do input log
newstart quit reset run test

 (buyer model)                 disagree   disagree   recommended
bid     price       agree       weakly    strongly     bid
*  b   10 one       0 a1       0 d1       9 d1+       0 one
0  u    0 two       0 a2       0 d2       0 d2+       0 two
0  y    0 three     0 a3       0 d3       0 d3+       7 three
0  e    0 four      0 a4       0 d4       0 d4+       0 four
0  r    0 five      0 a5       0 d5                   0 five
0       0 six       0 a6                              0 six

0  s    0 one
0  e    0 two
0  l    0 three
0  l    0 four
*  e   10 five
0  r    0 six
```
Figure 13 Seek quick agreement display

Figure 14 shows the state of the network after the buyer has increased the offer to $120 and the seller has reduced the price to $130. The active nodes are filled in.

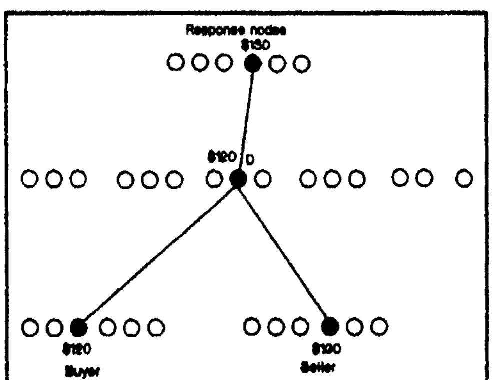

Figure 14 Buyer and seller differ by one dollar

The output screen is shown in Figure 15. The active nodes are underlined.

```
disp/ exam/ get/ save/ set/ clear cycle do input log
newstart quit reset run test
 (buyer model)                disagree   disagree   recommended
bid       price       agree     weakly     strongly     bid
0  b      0 one       0 a1      0 d1       0 d1+        0 one
0  u      0 two       0 a2      0 d2       0 d2+        0 two
*  y     10 three     0 a3      9 d3       0 d3+        0 three
0  e      0 four      0 a4      0 d4       0 d4+        9 four
0  r      0 five      0 a5      0 d5                    0 five
0         0 six       0 a6                              0 six

0  s      0 one
0  e      0 two
0  l      0 three
*  l     10 four
0  e      0 five
0  r      0 six
```
Figure 15 Buyer offers three and seller asks four dollars

The buyer and seller agree to settle at $130. This is depicted in Figure 16.

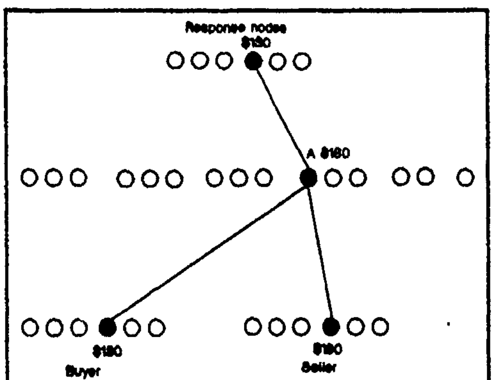

Figure 16 Buyer and seller agree

Figure 17 shows the screen display when both parties agree to $130.

```
disp/ exam/ get/ save/ set/ clear cycle do input log
newstart quit reset run test
 (buyer model)                disagree   disagree   recommended
bid      price      agree     weakly     strongly     bid

0   b    0 one      0 a1      0 d1       0 d1+        0 one
0   u    0 two      0 a2      0 d2       0 d2+        0 two
0   y    0 three    0 a3      0 d3       0 d3+        0 three
*   e   10 four     9 a4      0 d4       0 d4+        9 four
0   r    0 five     0 a5      0 d5                    0 five
0        0 six      0 a6                              0 six

0   s    0 one
0   e    0 two
0   l    0 three
*   l   10 four
0   e    0 five
0   r    0 six
```

Figure 17 Buyer and seller agree to four dollars

To test the strategy of conceding $10 when the buyer offer is $120 or more, the buyer offer will be set to $120 and the seller offer will be set to $140. Figure 18 shows this state of the network. The *120D+* and *130D+* nodes like the *100D+* and *110D+* nodes represent the idea that there is a $20 or more difference between the buyer and seller. The difference between these pairs of nodes (*120D+, 130D+* and *100D+, 110D+*) is that the buyer changes his strategy when currently offering $120 or more. Rather than increasing by $20, the buyer will now increase by only $10. This is an attempt to encourage the seller to decrease the asking price by indicating reluctance to increase the bid.

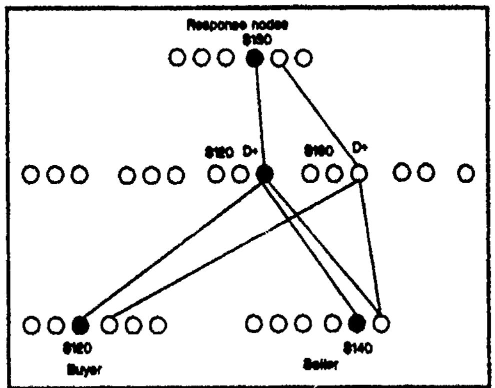

Figure 18 Concede only one dollar

V. BUILDING A NEURAL MODEL FOR A NEGOTIATION BASED ON TWO ISSUES

In this chapter, a neural network to support a negotiation based on two issues will be built. From the design standpoint, this neural network is very similar to the one for single issue negotiation described in the previous chapter. To describe the two issue negotiation model we use the bicycle buying problem again. In addition to the price issue, we will include the quality of a bicycle as a second issue. The method of building the network described in this chapter will follow the structure for building a network explained in chapter II.

A. DEFINE THE PROBLEM

1. The Buyer's Decision Making Process

This neural network will attempt to simulate the logical decision making processes of a buyer who is negotiating with a seller over the purchase price and quality of a bicycle. As in the single issue model, each time the seller makes an offer, the buyer will make a decision to accept the seller's offer or propose a new offer. Each offer will include two issues, price and quality. The price range is ($100 - $300). The quality range is (a,b,c), (a) being the best.

The decision strategy of the buyer is based on a comparison of two differences. The first difference is between the buyer's current price offer and the sellers current price offer. The second difference is between the buyer's current quality offer and the seller's current quality offer. These two differences will be compared to each other. The buyer will make a new offer based on a predetermined strategy based on this comparison.

2. **Representing the Buyer's Decision Making Process**

To create a set of schema which will reside in the mental model of the buyer, seven features of the negotiation environment from the buyer's perspective will be used. These features are:

- The buyer's current price and quality offer.
- The seller's current price and quality offer.
- A notion of the difference between the buyer's and seller's price offers.
- A notion of the difference between the buyer's and seller's quality offers.
- A comparison of the price and quality differences.
- The buyer's price and quality level response to the seller.
- The absence of a buyer's offer.

Each of these features contains a set of subfeatures. The *buyer's price and quality offer* feature consists of six subfeatures. Three of these subfeatures are discrete prices ($100, $200, $300). the remaining three subfeatures are discrete qualities (a,b,c). These subfeatures are the only possible prices and qualities which the buyer can offer to the seller. The *seller's price and quality offer* feature is the same as that of the buyer's.

It is assumed that both the buyer and the seller will always present both issues (price and quality) for negotiation simultaneously.

The *price difference* feature will consist of six subfeatures. Each of these subfeatures is the difference between the buyer's price offer and the seller's price offer. These subfeatures are:

1. The seller agrees with the buyer's $100 offer.
2. The seller is asking $200 and the buyer is offering $100.
3. The seller is asking $300 and the buyer is offering $100.
4. The seller agrees with the buyer's $200 offer.
5. The seller is asking $300 and the buyer is offering $200.
6. The seller agrees with the buyer's $300 offer.

This *price difference* feature is necessary because the buyer's decision making strategy depends on recognizing the difference in price offered.

The *quality difference* feature is similar to the price difference feature. There are six subfeatures representing the difference between the buyer's quality offer and the seller's quality offer. These subfeatures are:

1. The seller agrees with the buyer's "a" quality offer.
2. The seller is asking "b" quality and the buyer is offering "a" quality.
3. The seller is asking "c" quality and the buyer is offering "a" quality.
4. The seller agrees with the buyer's "b" quality offer.
5. The seller is asking "c" quality and the buyer is offering "b" quality.
6. The seller agrees with the buyer's "c" quality offer.

This *quality difference* feature is necessary because the decision making strategy of the buyer depends on recognizing the difference in quality offered.

There are thirty six subfeatures within the *price/quality difference comparison* feature. The subfeatures represent the idea of comparing the price difference feature to the quality difference feature. Since there are six subfeatures in each of the price and quality features, thirty six comparisons are necessary (6 * 6). For example, the price of $100 dollars may be offered by both buyer and seller but the seller is offering quality "c" while the buyer is offering quality "a", the difference in price (none) and the difference in quality (disagree strongly) would comprise a single comparison subfeature, (price/quality). The strategy of the buyer is based on the results of this comparison.

The *buyer response* feature is composed of six subfeatures. There are three price ($100, $200, $300) and three quality (a,b,c) subfeatures. This feature is not divided into separate price and quality features since it is assumed that the buyer will always present both issues (price and quality) for negotiation simultaneously.

The feature representing the *absence of a buyer offer* has a single subfeature - the absence of a buyer offer. This feature represents a special case of the buyer with no current offer. When the buyer has no current offer, no difference in offers will exist and no comparison of differences may be done. This feature will bypass the use of the difference and comparison features to allow an offer to be made.

Figure 19 is a representation of the mind of the buyer. This figure assumes that the buyer and seller have each made an offer. The buyer will "see" her price and quality offer and the seller's price and quality offer. The buyer will then recognize more of the picture by "seeing" the differences between her price and the seller's price and "seeing" the difference between her quality and that of the seller. A larger view of the picture is formed when the buyer "sees" a comparison of the price difference and the quality difference. The complete picture is formed when the buyer "sees" the response offer which will be communicated to the seller.

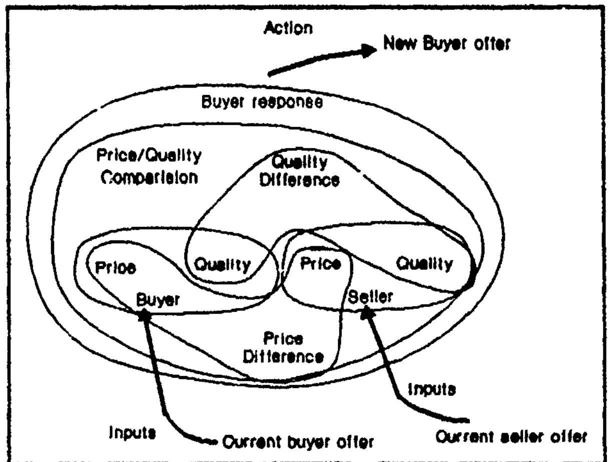

Figure 19 Buyer's mental model of negotiation

B. CHOOSE THE PARADIGM

As for the single issue negotiation, the constraint satisfaction paradigm is chosen for the two issue negotiation model.

C. CONFIGURE THE NETWORK

As discussed, two issues will be negotiated, price and quality. The buyer will consider any possible combination of price and quality as acceptable but will prefer to negotiate with the seller in an attempt to pay the lowest price ($100) and obtain the highest quality (a).

The buyer has four objectives. They are:

1 Obtain agreement on both issues

2 Accept an agreement on at least one issue

3 Always concede one unit of quality before price

4 Do not seek the poorest quality for the best price

The model has four strategies built into it. They are:

A If there is no current buyer offer, offer lowest price ($100) and highest quality (a).

B If both buyer and seller agree on an issue, the outcome of that issue will not be subject to change.

C The buyer will concede on quality but not on price. If the current buyer offer of price and quality are $100,a or $200,b and the seller disagrees on both issues, the buyer will respond with $100,b if current buyer offer is $100,a and the seller disagrees with both price and quality.

D If the current buyer offer of quality is one level lower than price (ie. $100,b) and the seller disagrees with both price and quality, concede $100 (ie. respond with $200,b if current buyer offer is $100,b and the seller disagrees with both price and quality).

Strategy A is a default starting strategy and is self explanatory. Strategy B and objective 2 assume that both parties agree to a rule at the beginning of negotiation that once agreement is reached on a single issue, the agreed upon value will not be subject to change.

Strategy C reflects the buyer preference that a lower quality will be acceptable for in exchange for a more favorable price (objective 3). The buyer will not pay more to accept a higher quality unless the quality has already been agreed upon according to the rule associated with strategy B.

Strategy D reflects the idea that the buyer does not desire to pay the lowest price ($100) and obtain the worst quality (c). If the seller is offering the highest price and lowest quality ($300, c), the buyer would prefer to pay $200 in an attempt to induce the seller to split the difference of the original offers (assume that the buyer's original offer is $100,a and the seller's original offer is $300,c) and accept the buyer's offer of $200 and b quality.

Each of these strategies result in a buyer action on the negotiation environment. The application of the strategies are a result of how the buyer interprets the environment.

To have the model correctly interpret the environment, seven features will be needed. These features of the negotiation environment were described in the problem definition phase. These features are:

- The buyer's current price and quality offer.
- The seller's current price and quality offer.
- A notion of the difference between the buyer's and seller's price offers.
- A notion of the difference between the buyer's and seller's quality offers.
- A comparison of the price and quality differences.
- The buyer's price and quality level response to the seller.
- The absence of a buyer offer.

Each feature will be represented by a set of nodes (hypotheses) and connections (constraints). A node will have a continuous range of activity from zero to one. Zero activity corresponds to a node being inactive (the hypothesis is false).

The buyer price and quality offer feature will be represented by six nodes and connections. Three nodes correspond to the buyer's price offer and three correspond to the buyer's quality offer. The connections to these nodes are the input connections which convey the current buyer's price and quality offer information. The seller's price and quality offer feature will be similarly represented. Both of these features are shown in Figure 20.

For this negotiation, it is assumed that the $100 offer is the buyer's best price alternative and the $300 offer is the buyer's worst price alternative. Quality "a" is the best quality and quality "c" is the worst quality. The buyer's highest preference is to pay the lowest price ($100) and obtain the highest quality ("a"). If necessary, the buyer is willing to pay $300 and accept the lowest quality.

The connections shown in Figure 20 to these nodes convey the seller's current price and quality offer information from the negotiation environment.

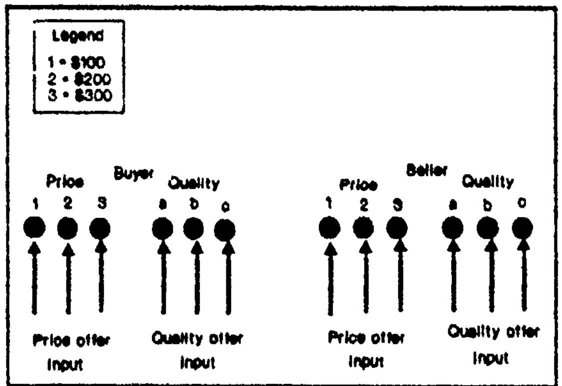

Figure 20 Buyer and seller input features

The buyer will need to determine the difference in price offers. This difference is the idea that the seller agrees to the buyer's offer, disagrees or disagrees significantly when the buyer is offering $100. If the buyer is offering $200 the seller could only agree or disagree. If the buyer is offering $300 the seller may only agree. The limited number of price differences are due to the assumption that the seller will not offer a lower price than the buyer is offering.

Figure 21 shows the difference feature as represented in the neural network. Each buyer price is compared with the seller price which equals or exceeds it. The A, D and D+ symbols above the nodes indicate the meaning of the node. The *$100 difference* A (Agree) node indicates that the buyer and seller agree to a price of $100. The *$100 difference D* (Disagree) node indicates that the seller is asking $200 while the buyer is offering $100. The *$100 difference D+* (Disagree strongly) node indicates that the seller is asking $300 and the buyer is offering $100.

It is important to bear in mind that a set of nodes and connections <u>and</u> the state of activity of the nodes are necessary to represent a feature. The idea of agreement at $100 only exists when the buyer's *$100 price offer* node, the seller's *$100 price offer* node and the *$100 difference A* (Agreement) node are active.

Each of the price difference nodes has an inhibitory connection to the other price difference nodes. This type of connection is shown in the Figure 21 by a curved line with a solid dot at each end (this convention of representing inhibitory connections will prevail in each figure).

In Figure 21, only a sample of the inhibitory connections are displayed for clarity. The inhibitory connection prevents conflicting nodes from becoming active. Only a single price difference node should be active at any point in time.

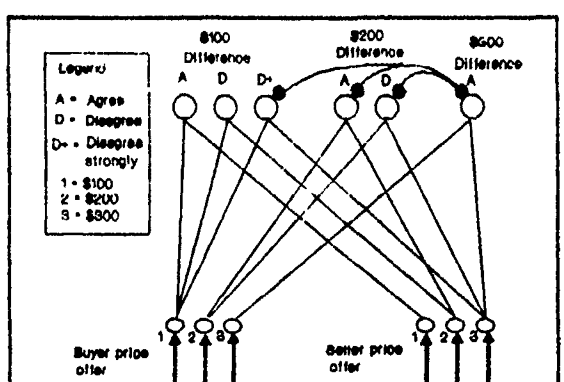

Figure 21 Price difference feature

The buyer will need to determine the difference in quality offers. The quality difference is the idea that the seller agrees to the buyer's quality offer, disagrees or disagrees strongly. This difference feature has the same architecture as the price feature. It is assumed that the seller will not offer a lower quality than the buyer is offering.

Figure 22 shows how the quality difference feature is represented in the neural network. Each buyer quality offer is compared with the seller quality offer which equals or exceeds it. The A (Agree), D (Disagree) and D+ (Disagree strongly) symbols above the nodes indicate the meaning of the node. The *quality "a" difference* A (Agree) node indicates that the buyer and seller agree to quality "a". The *quality "a" difference D* (Disagree) node indicates that the buyer is offering quality "a" while the seller is offering quality "b". The *quality "a" difference D+* (Disagree strongly) node indicates that the buyer is offering quality "a" and the seller is offering quality "c".

Each of the quality difference nodes has an inhibitory connection to the other quality difference nodes. This prevents conflicting nodes from becoming active. Only a single quality difference node should be active at any point in time.

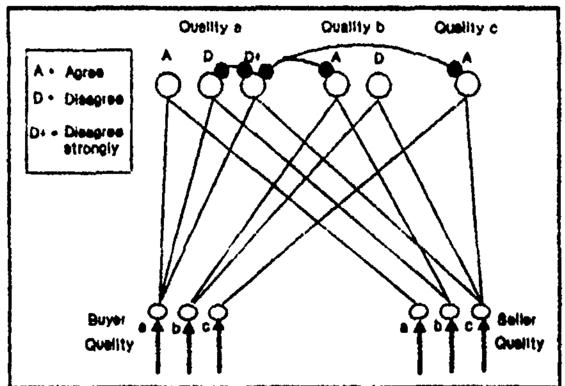

Figure 22 Quality difference feature

The price/quality comparison feature is shown in Figure 23. There are thirty six nodes in this feature. Only twelve of these nodes and their connections are shown in this figure to preserve clarity.

Each individual comparison node represents a pairwise comparison of the price difference feature and the quality difference feature. Each price difference and quality difference node connect with six comparison nodes. An inhibitory connection exists between each of the comparison nodes which will allow only a single node to become active.

For convention, each comparison node will be labeled according to the difference nodes which are connected to it (ie. price difference, quality difference). The *P2D,QaA* node would represent the comparison node which is connected to the *$200 difference D* (Disagree) difference node and the *quality "a" difference A* (Agree) difference node.

The *P1A,QaA* comparison node represents the idea that the seller agrees to the buyer's price offer of \$100 <u>and</u> agrees to the buyer's quality "a" offer. This situation would be highly advantageous to the buyer. The *P1D,QaA* comparison node represents the idea that the seller disagrees with the buyer's \$100 offer <u>and</u> that the seller agrees with the buyer's offer of quality "a".

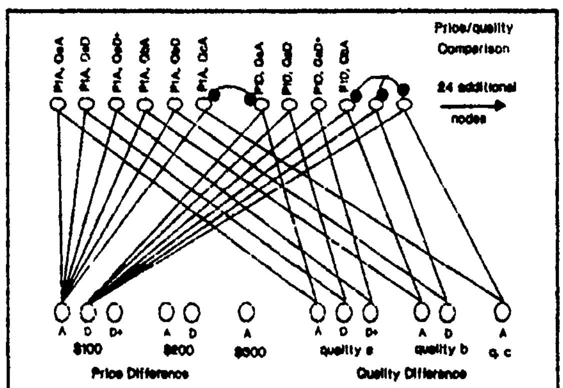

Figure 23 Price/quality comparison feature

After the buyer has determined the difference in price and quality offers and has made a comparison of these differences, the buyer will develop a new offer. This offer is the result of the mental picture which has been formed in the previous feature sets and the buyer's strategy on how to respond to each picture.

For example, strategy C prescribes that if the current buyer's price and quality offer are at the same level (ie. \$100,a or \$200,b) and the seller disagrees on both issues, the buyer will concede on quality but not on price. To implement this strategy, the comparison nodes (*P1D,QaD*), (*P1D,QaD+*), (*P1D+,QaD*) and (*P1D+,QaD+*) would each be connected to the *$100* *price* response node and the *quality b* response node. If these comparison nodes became active (indicating that their associated hypothesis is true), they would activate these response nodes ($100,b).

Figure 24 shows the connections between the response nodes and twelve of the comparison nodes.

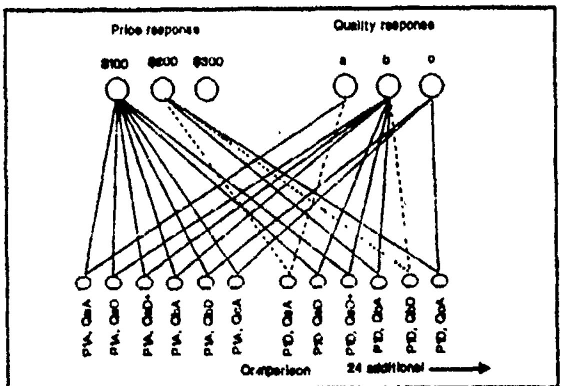

Figure 24 Response feature

The dashed line connecting the *P1D,QaA* node to the *$200* and *quality* "a" response nodes implements strategy B. Strategy B follows the predetermined rule that if an issue is agreed on, the agreed upon value will not change. The price response will be $200 and the quality response will be "a". The *P1D,QaA* node indicates that both parties have agreed to quality "a" but disagree on the buyer's price offer of $100. If this node is active, the neural network will respond with a price offer of $200 and quality "a".

The dashed line connecting the *P1D,QbD* node to the *$200* response and the *quality "b"* response nodes reflects strategy D. Strategy D prescribes that if the current buyer offer of quality is one level lower than price (ie. $100,b) and the seller disagrees with both price and quality, the buyer will concede $100.

The *P1D,QbD* node indicates that the seller disagrees with the buyer's $100 offer and with the buyer's "b" quality offer. According to the buyer's strategy, an offer of $200 and "b" quality should be made.

There is a special case where the buyer does not have a current offer. This feature of the negotiation is represented in Figure 25. If the buyer does not have a current offer, an offer of $100 and quality "a" will be made. The inhibitory connections between the buyer price and quality offer nodes and the no offer node will act to turn this feature off if an offer is present. The no offer node is biased to be slightly on.

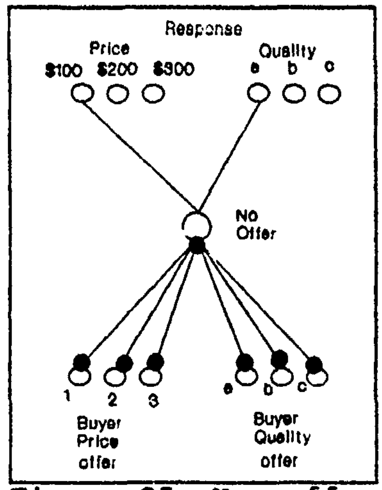

Figure 25 No offer feature

Figure 26 shows the complete network architecture. For clarity, the thirty six comparison nodes are represented by an ellipse and most of the connections are not displayed.

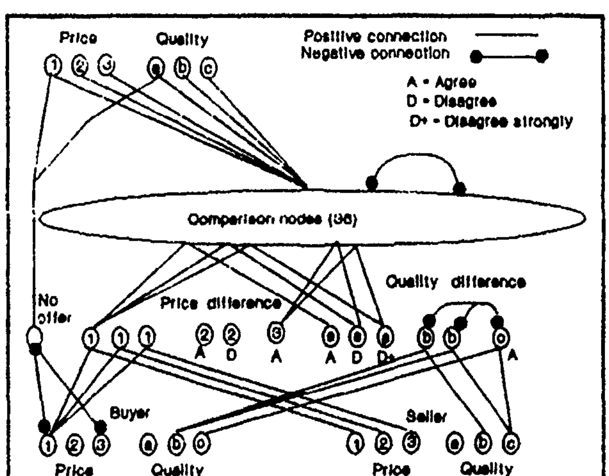

Figure 26 Network architecture

Figure 27 and Figure 28 show the two issue network description file which is used by the CS program to configure the neural network. Figure 27 shows the description of the first twenty four nodes. Figure 28 shows the description of the remaining forty three nodes and the biases assigned to each node.

The structure for this file is the same as for the single issue network description file. Each row indicates the existence of a node and the connection weights to that node. Row one, two and three are the *buyer's current price offers* and row four five and six are the *buyer's current quality offers*. Row seven through nine represent the *seller's current price offer*, ten through twelve represent the seller's *current quality offer*. Each of the current price offer nodes are connected with a weight of +0.5 to the *price difference* nodes represented in row thirteen through eighteen. Each current quality offer node is connected a weight of +0.5 to the *quality difference* nodes with in row nineteen through twenty four. All of these nodes (offer and difference) are assigned a bias of -0.7.

The connection weight of +0.5 and bias of -0.7 was chosen so that two offer nodes would have to be active in order to activate the one node which they are both connected to. A single offer node will not be able to activate a difference node and a single difference node will not be able to activate an offer node with these connection weights and biases. This will prevent a "rebound" effect where the upper layer (in this case the difference nodes) could erroneously activate a lower layer node (an offer node).

The small "x" in the buyer's price and quality nodes represents a negative connection of -1.0 to the *no offer* node. This inhibitory connection is needed to ensure that the no offer node does not become active when a buyer offer is present.

Each *price difference* node inhibits al other *price difference* nodes as indicated by the block of x's in row thirteen through eighteen. Each *quality difference* node inhibits all other *quality difference* nodes as indicated by the block of x's in row nineteen through twenty four. This inhibition is necessary since the difference nodes are connected to the offer nodes and the comparison nodes.

The combined weighted input ($\Sigma w_{ij}a_j$) to a difference node which is connected to one active offer node and one active comparison node is +0.75 (0.5*1 + 0.25*1). This input will exceed the -0.7 bias assigned to the difference node.

Since the state of activation of the nodes in the network will generally spread from the offer to the difference and then to the comparison nodes before any "rebound" could occur, the inhibitory connection in the difference nodes will take effect before the "rebound" occurs and prevent "rebound".

The a,b and c in the first twelve rows of the price and quality difference nodes show the symmetrical connections to the buyer and seller price and quality input nodes which must exist for the CS network.

The *price difference* nodes and the *quality difference* nodes are positively connected (+0.25) to the *comparison* nodes. This connection is indicated by the "d". The *comparison* nodes are assigned a bias of -0.4. This negative bias prevents a comparison node from becoming active if it is connected to a single active difference node and a single active response node. It is not sufficient to prevent a comparison node from becoming active if it is connected to a single active difference node and two active response nodes. In this case the weighted input ($\Sigma w_{ij}a_j$) will be +0.45 (0.25*1 + 0.1*1 + 0.1*1). To prevent a "rebound" effect an inhibitory connection of -1.0 between each comparison node will be used. The x's associated with each *comparison* node represent the inhibitory connection to all other *comparison* nodes.

The *comparison* nodes are divided into two sections. The first section considers that at least one difference node represents a disagreement in either price or quality. The second section considers agreement on both issues.

Each *comparison* node is symmetrically connected with the difference node connected to it as indicated by the "d" in the thirteenth through twenty fourth column of each *comparison* node. The "z" associated with each *comparison* node represents the connection weight (+0.1) of that *comparison* node with a *response* node. Each *comparison* node is connected with two response nodes. One connection is with a price response and the other connection is with a quality response.

Each *response* node is symmetrically connected with a *comparison* node as indicated by the presence of a "¿" in columns twenty five through fifty one and in columns fifty nine through sixty seven. No inhibitory connection is needed between the *response* nodes and no bias is assigned to the response nodes since all "rebound" effect has been removed from the network.

The "a" in the $100 price response node and the quality "a" response node represents the connection these nodes have with the *no offer* node.

The connections to the *no offer* node are in row fifty eight. This node is negatively connected to the *buyer price offer* and *buyer quality offer* input nodes as indicated by the x's in the first six columns. It is also connected to the *$100 response* and *quality "a" response* nodes, indicated by the a's in column fifty two and fifty five. This node is biased to be slightly active (+0.1).

Activity in the input nodes will deactivate this node. If there is no buyer offer, this node will activate the response nodes it is connected to. The bias weight and connection weight were chosen so that the effect cf this node on the response nodes would not be significant until the network has updated the activation states of all of the other nodes. It is possible for this node to be the first node to have it's activation state updated (random node selection for update). If this node were updated and then both of the response nodes connected to the no offer node had their activation state updated prior to any other node <u>and</u> the no offer node caused the response nodes to immediately become fully active, the network could behave erratically.

The biases associated with each node are indicated in the bias section of the file shown in Figure 28.

```
definitions:
nunits 67
end
constraints:
a 0.5
b 0.5
c 0.5
d 0.25
z 0.1
x -1.0
s -0.7
w -0.4
t 0.1
end
network:
                   Buyer Input nodes
............abc..........................................X.........
...............ab........................................X.........
.................a.......................................X.........
..................abc....................................X.........
.....................ab.............................   ...X.........
.......................a.............................  . X.........
                   Seller input nodes
............a......................................................
.............b.a...................................................
..............c.ba.................................................
..................a................................................
...................b.a.............................................
....................c.ba...........................................
                   Price difference nodes
a.....a......XXXXX......ddd...............................ddd.  ....
b......b....X.XXXX.........dddddd...........................    ....
c.......c...XX.XXX...............dddddd............................
.a.....a....XXX.XX.....................ddd...................ddd...
.b......b...XXXX.X........................dddddd...................
..a.....a...XXXXX...............................ddd.............ddd
                   Quality difference nodes
...a.....a.........XXXXX...d.....d........d...............d..d..d..
...b......b.......X.XXXXd...d.....d....d...d....d..................
...c.......c......XX.XXX.d...d.....d....d...d....d.................
....a.....a.......XXX.XX......d.....d........d.............d..d..d.
....b......b......XXXX.X..d....d.....d...d....d...d................
.....a.....a......XXXXX.........d.....d........d............d..d..d
```

Figure 27 Network description file part one

```
                         Comparison nodes
............d......d.....xxxxxxxxxxxxxxxxxxxxxxxxxxx...z..xxxxxxxxx
............d.......d...x.xxxxxxxxxxxxxxxxxxxxxxxxxx...z..xxxxxxxxx
............d.........d.xx.xxxxxxxxxxxxxxxxxxxxxxxxx....z.xxxxxxxxx
.............d....d.....xxx.xxxxxxxxxxxxxxxxxxxxxxx.z.z...xxxxxxxxx
.............d.....d....xxxx.xxxxxxxxxxxxxxxxxxxxxxx...z..xxxxxxxxx
.............d......d...xxxxx.xxxxxxxxxxxxxxxxxxxxxx...z..xxxxxxxxx
.............d.......d..xxxxxx.xxxxxxxxxxxxxxxxxxxx.z..z..xxxxxxxxx
.............d........d.xxxxxxx.xxxxxxxxxxxxxxxxxxx.z..z..xxxxxxxxx
.............d.........dxxxxxxxx.xxxxxxxxxxxxxxxxxx.z...z.xxxxxxxxx
..............d...d.....xxxxxxxxx.xxxxxxxxxxxxxxxxx.z.z...xxxxxxxxx
..............d....d....xxxxxxxxxx.xxxxxxxxxxxxxxxxx...z..xxxxxxxxx
..............d.....d...xxxxxxxxxxx.xxxxxxxxxxxxxxxx...z..xxxxxxxxx
..............d......d..xxxxxxxxxxxx.xxxxxxxxxxxxxx.z..z..xxxxxxxxx
..............d.......d.xxxxxxxxxxxxx.xxxxxxxxxxxxx.z..z..xxxxxxxxx
..............d........dxxxxxxxxxxxxxx.xxxxxxxxxxxx.z...z.xxxxxxxxx
...............d...d....xxxxxxxxxxxxxxx.xxxxxxxxxxx.z..z..xxxxxxxxx
...............d....d...xxxxxxxxxxxxxxxx.xxxxxxxxxx.z..z..xxxxxxxxx
...............d......d.xxxxxxxxxxxxxxxxx.xxxxxxxxx.z...z.xxxxxxxxx
................d.d.....xxxxxxxxxxxxxxxxxx.xxxxxxxx..zz...xxxxxxxxx
................d..d....xxxxxxxxxxxxxxxxxxx.xxxxxxx.z..z..xxxxxxxxx
................d...d...xxxxxxxxxxxxxxxxxxxx.xxxxxx.z..z..xxxxxxxxx
................d....d..xxxxxxxxxxxxxxxxxxxxx.xxxxx..z.z..xxxxxxxxx
................d.....d.xxxxxxxxxxxxxxxxxxxxxx.xxxx.z...z.xxxxxxxxx
................d......dxxxxxxxxxxxxxxxxxxxxxxx.xxx..z..z.xxxxxxxxx
.................d.d....xxxxxxxxxxxxxxxxxxxxxxxx.xx..z.z..xxxxxxxxx
.................d..d...xxxxxxxxxxxxxxxxxxxxxxxxx.x..z.z..xxxxxxxxx
.................d....d.xxxxxxxxxxxxxxxxxxxxxxxxxx...z..z.xxxxxxxxx
                         Response nodes
........................zzz.zz....zz...................azzz......
...........................z..zzzz..zzzzzz.zz.z............zzz...
..........................................z..z.zzzz..............zzz
...........................z.z...z........z...............az..z..z..
........................zz..zzzz..zzzz.zz..zzz..zz.........z..z..z.
..........................z....z....z..z...zz..z..........z..z..z
                         No offer node
xxxxxx...............................................a..a...........
                         Comparison nodes where an agreement exists for both issues
............d.....d.....xxxxxxxxxxxxxxxxxxxxxxxxxxxxx..z....xxxxxxxx
............d........d..xxxxxxxxxxxxxxxxxxxxxxxxxxxxx...z..x.xxxxxxx
............d..........dxxxxxxxxxxxxxxxxxxxxxxxxxxxxx....z.xx.xxxxxx
...............d..d.....xxxxxxxxxxxxxxxxxxxxxxxxxxxx.z.z...xxx.xxxxx
...............d....d...xxxxxxxxxxxxxxxxxxxxxxxxxxxx.z..z..xxxx.xxxx
...............d.......dxxxxxxxxxxxxxxxxxxxxxxxxxxxx.z...z.xxxxx.xxx
.................dd.....xxxxxxxxxxxxxxxxxxxxxxxxxxxx..zz...xxxxxx.xx
.................d..d...xxxxxxxxxxxxxxxxxxxxxxxxxxxx..z.z..xxxxxxx.x
.................d.....dxxxxxxxxxxxxxxxxxxxxxxxxxxxx..z..z.xxxxxxxx.
end
biases:
wwwwwwwwwwwwwwwwwwwwwwwwvvvvvvvvvvvvvvvvvvvvvvvvvvvv.....tvvvvvvvvv
end
```

Figure 28 Network description file part two

D. TEST THE NETWORK

Testing the network will consist of simulating a negotiation. This is by no means an exhaustive test of the validity of the neural network but serves only to show how the network functions.

In Figure 29 the nodes and connections which will be active are shown as filled in circles. None of the buyer input nodes are active. The no offer feature node is active. The no offer node will activate the $100 and quality "a" response nodes. This is the initial offer that the buyer will convey to the seller.

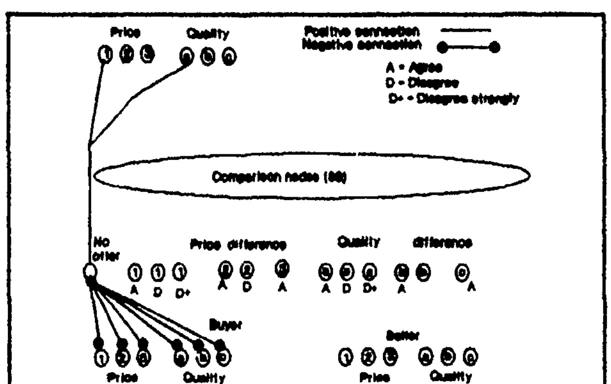

Figure 29 Initial buyer offer

Figure 30 shows the screen display of this situation. The buyer and seller input nodes are displayed at the left of the screen under the "INPUT" heading. The "P" indicates the nodes associated with the buyer price input. "Q" indicates the nodes associated with the buyer's quality input. "p" indicates the seller's price input and "q"indicates the seller's quality input. In this case, there is no input to the network. The buyer response is displayed on the right side of the display under the heading of "RESPONSE". P1 indicates the $100 price node and Qa indicates the quality "a" node. The 9 in front of each of these nodes indicates that these nodes are active with an activity level of 0.9 in a range from zero to one. The "NO" node in the lower center portion of the screen is the no offer node. This node is fully active (1.0) which is indicated by the "*".

```
disp/ exam/ get/ save/ set/ clear cycle do input log newstart quit
reset run test

 INPUT                                                                RESPONSE
  0 0 P1  0 A1  0 D1  0 D+1                                            9 P1
B 0 0 P2  0 A2  0 D2        0 ADa 0 AD+ 0 ADb                          0 P2
U 0 0 P3  0 A3                                                         0 P3
Y                           0 DAa 0 DDa 0 DD+ 0 DAb 0 DDb 0 DAc
E 0 0 Qa  0 Aa  0 Da  0 D+a
R 0 0 Qb  0 Ab  0 Db        0 +Aa 0 +Da 0 +D+ 0 +Ab 0 +Db 0 +Ac
  0 0 Qc  0 Ac                                                         9 Qa
                            0 ADa 0 AD+ 0 ADb                          0 Qb
S 0 0 p1                                                               0 Qc
E 0 0 p2                    0 DAa 0 DDa 0 DD+ 0 DAb 0 DDb 0 DAc
L 0 0 p3
L                           0 ADa 0 AD+ 0 ADb
E 0 0 qa
R 0 0 qb                                * NO
  0 0 qc        0 A1a 0 A1b 0 A1c 0 A2a 0 A2b 0 A2c 0 A3a 0 A3b 0 A3c
```

Figure 30 Screen display when buyer starts negotiation

Figure 31 shows the state of the network when the buyer's current offer is $100 and quality "a" and the seller's current offer is $300 and quality "c".

The $100 D+ difference node is active indicating that the seller strongly disagrees with the buyer's $100 offer. The quality "a" D+ difference node is active indicating that the seller strongly disagrees with the buyer's quality "a" offer. Both of these nodes will activate a single node in the comparison feature level. The sum of two active difference nodes is necessary to activate a comparison node. A single difference node will not be able to activate a comparison node.

The activated comparison node (in this case it will be the P1D+,QaD+ node) will activate the $100 price response and the quality "b" quality response nodes. This reflects the strategy of the buyer by conceding on quality but not on price.

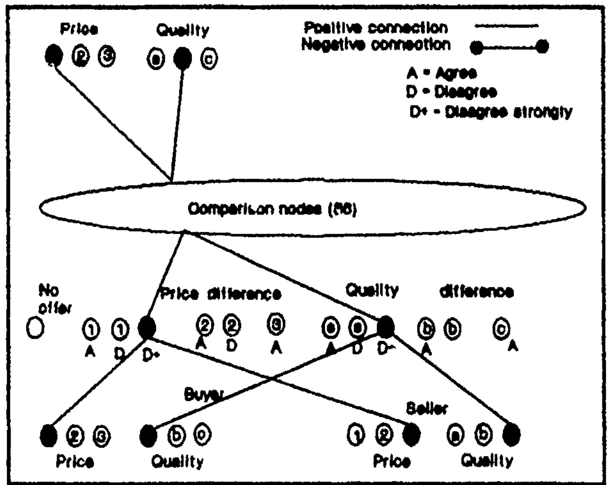

Figure 31 Strong disagreement with buyer offer

Figure 32 shows the screen display associated with the strong disagreement situation. The buyer and seller inputs are indicated by a "*" next to their input nodes in the "INPUT" portion of the screen. Display size limitations allow only three characters for identification of each node. The D+1 node is the $100 strong disagreement node. This node is in a wedge shaped group of six nodes, all of which are the price disagreement nodes. the A1 node in this group represents the $100 agreement node, the D1 node represents the $100 disagreement node.

The second wedge shaped group of six nodes are the quality difference nodes. The D+a node is the quality "a" strong disagreement node. The Ab node is the quality "b" disagreement node. In this screen, both strong disagreement nodes are active as indicated by the "*" to the left of each.

The large group in the center of the screen represents the comparison nodes which are connected to at least one difference node. The display convention for these nodes is that the first row is connected to the $100 agreement node. Row two is connected to the $100 disagreement node. Row three is connected to the $100 strong disagreement node. Row four is connected to the $200 agreement node. Row five is connected to the $200 disagreement node and row six is connected to the $300 agreement node.

Within each row the connections to the quality difference nodes is indicated by the last two characters of each node. For example, the ADa node is connected to the $100 A (agree) price difference node and to the quality "a" D (Disagree) quality difference node. The DAb comparison node in row two is connected to the $100 D (disagree) price difference node and the quality "b" A (agree) quality difference node. In this screen the +D+ node in row three is active indicating that there is strong disagreement with the buyer's $100 offer <u>and</u> with the buyer's quality "a" offer.

The response nodes, P1 and Qb are activated indicating that the buyer's response should be to offer $100 and quality "b" to the seller.

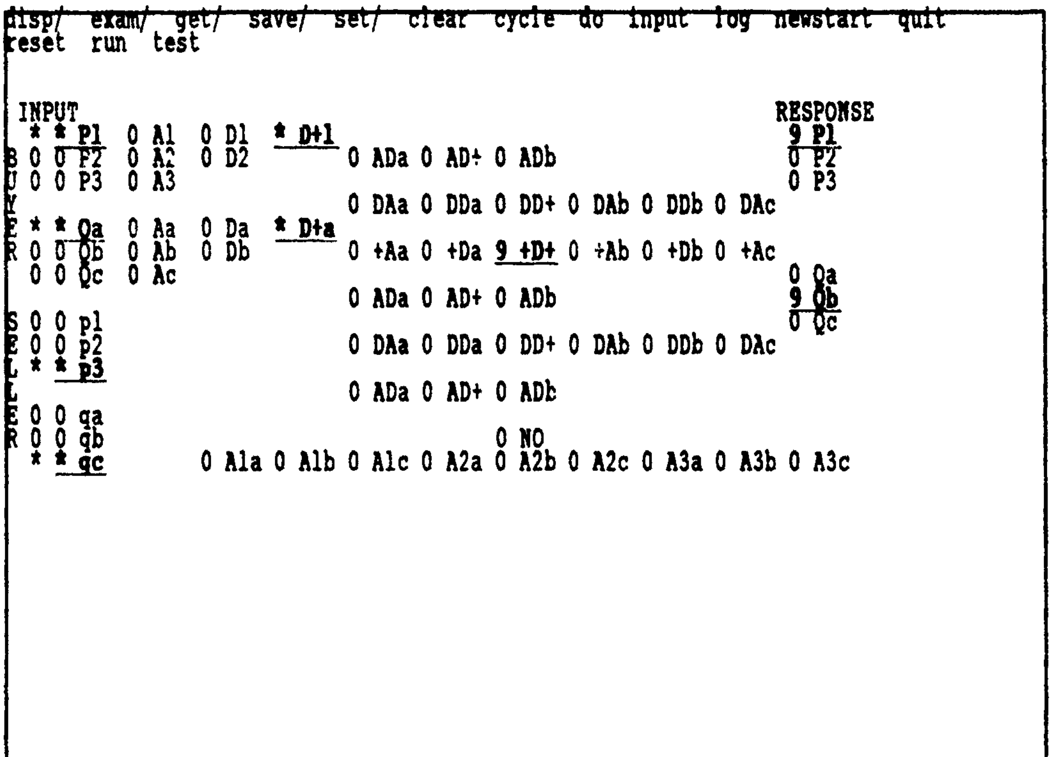

Figure 32 Screen display for strong difference with buyer offer

Figure 33 shows the state of the network when the buyer is currently offering \$100 and quality "b" and the seller is asking \$200 and quality "c". The network will develop a response in this case according to strategy D. Strategy D prescribes that if the current buyer offer of quality is one level lower than price (ie. \$100,b) and the seller disagrees with both price and quality, concede \$100. The response generated will be for the buyer to offer \$200 and quality "b".

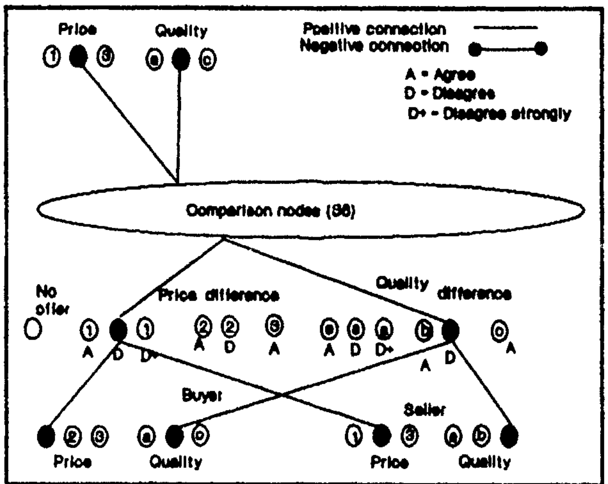

Figure 33 Buyer and seller disagree

Figure 34 shows the screen display associated with the situation described above. The D1 price difference node is active and the Db quality difference nod• is active. The comparison node DDb is active indicating that there is disagreement with the $100 buyer price offer <u>and</u> the quality "b" buyer quality offer. The response nodes P2 and Qb are active indicating that the buyer response is to offer $200 and quality "b".

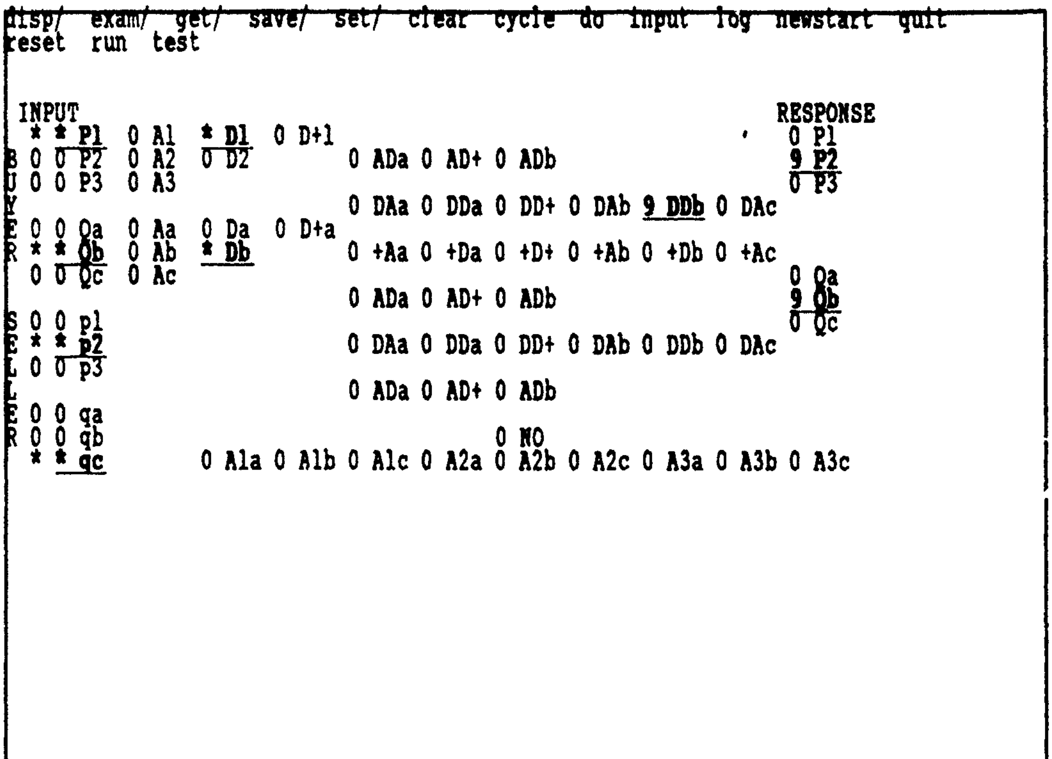

Figure 34 Screen display of strategy D

Figure 35 shows the state of the network when the buyer has offered $200 and quality "b". The seller has asked for $200 and quality "c". The buyer will concede to the seller demand with an offer of $200 and quality "c". The buyer cannot change the price offer of $200 since this issue has been agreed on as indicated by the $200 A (agree) price difference node.

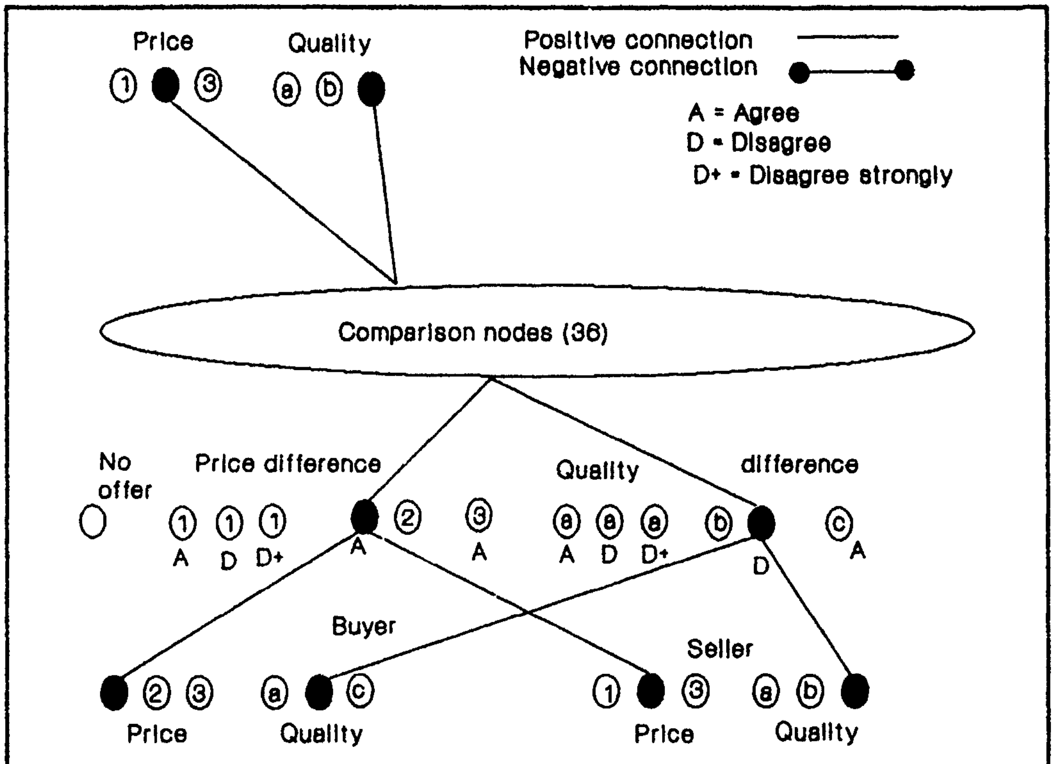

Figure 35 Buyer meets seller demand

Figure 36 shows the display of the price agreement, quality disagreement situation. The A2 price difference node and the Db quality difference nodes are active. The ADb comparison node is active indicating that the price of $200 is agreed <u>and</u> that the seller disagrees with the buyer's quality "b" offer. The response nodes P2 and Qc are active indicating that the buyer response should be $200 and quality "c". This will meet the demand of the seller and the negotiation will end.

This result seems to make sense. As defined in the strategy, the buyer was cooperative and the seller ultimately took advantage of the buyer.

```
disp/ exam/ get/ save/ set/ clear cycle do input log newstart quit
reset  run  test

 INPUT                                                           RESPONSE
  0 0 P1   0 A1  0 D1  0 D+1                                      0 P1
B * * P2   * A2  0 D2        0 ADa 0 AD+ 0 ADb                    9 P2
U 0 0 P3   0 A3                                                   0 P3
Y                            0 DAa 0 DDa 0 DD+ 0 DAb 0 DDb 0 DAc
E 0 0 Qa   0 Aa  0 Da  0 D+a
R * * Qb   0 Ab  * Db        0 +Aa 0 +Da 0 +D+ 0 +Ab 0 +Db 0 +Ac
  0 0 Qc   0 Ac                                                    0 Qa
                             0 ADa 0 AD+ 9 ADb                     0 Qb
S 0 0 p1                                                           9 Qc
E * * p2                     0 DAa 0 DDa 0 DD+ 0 DAb 0 DDb 0 DAc
L 0 0 p3
L                            0 ADa 0 AD+ 0 ADb
E 0 0 qa
R 0 0 qb                                 0 NO
  * * qc         0 A1a 0 A1b 0 A1c 0 A2a 0 A2b 0 A2c 0 A3a 0 A3b 0 A3c
```

Figure 36 Agreement on two dollars disagreement on quality

The screen display indicating of the final state of negotiation is displayed in Figure 37. The price and quality difference nodes indicating agreement are active (A2 and Ac). The A2c node in the bottom row is active indicating that a price of $200 <u>and</u> quality "c" have been agreed upon. The response nodes P2 and Qc reflect the final offer.

```
disp/  exam/  get/  save/  set/  clear  cycle  do  input  log  newstart  quit
reset  run  test


 INPUT                                                                   RESPONSE
  0 0 P1   0 A1   0 D1   0 D+1                                           0 P1
B * * P2   * A2   0 D2        0 ADa 0 AD+ 0 ADb                          9 P2
U 0 0 P3   0 A3                                                          0 P3
Y                             0 DAa 0 DDa 0 DD+ 0 DAb 0 DDb 0 DAc
E 0 0 Qa   0 Aa   0 Da   0 D+a
R 0 0 Qb   0 Ab   0 Db        0 +Aa 0 +Da 0 +D+ 0 +Ab 0 +Db 0 +Ac
  * * Qc   * Ac                                                          0 Qa
                              0 ADa 0 AD+ 0 ADb                          0 Qb
S 0 0 p1                                                                 9 Qc
E * * p2                      0 DAa 0 DDa 0 DD+ 0 DAb 0 DDb 0 DAc
L 0 0 p3
L                             0 ADa 0 AD+ 0 ADb
E 0 0 qa
R 0 0 qb                                  0 NO
  * * qc         0 A1a 0 A1b 0 A1c 0 A2a 0 A2b 9 A2c 0 A3a 0 A3b 0 A3c
```

Figure 37 Both issues are agreed upon

V. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

A. CONCLUSIONS

The neural networks developed in this thesis have demonstrated the ability to mimic some simple thought processes of a negotiator. We were able to model a bilateral negotiation from the point of view of the buyer; the seller being the opponent. Chapter IV used the price of the good (i.e., the bicycle) as the only bargaining issue to facilitate the explanation of the network development process. In Chapter V, we introduced a second issue -- i.e., the quality of the good -- to demonstrate the ability of the model to address a more life-like negotiation. The idea of Chapter V was that if a two-issue problem could be built, an n-issue problem could also be implemented.

As a final remark, the two neural networks proposed in this thesis exhibit a behavior very similar to that of an expert system. The major difference between the neural networks and an expert system lies in the way information is represented. Facts and knowledge can be represented by rules in an expert system. They are represented by nodes and connections in a neural network. The process of evaluating why a decision is made can be done in an expert system by having it reveal which rules were invoked during the consultation. In the neural network each node carries its meaning explicitly; a method of reasoning can be readily seen by observing which nodes are active.

The findings of this thesis suggest that continued research in neural networks to model the thought processes of negotiators holds great promise. The value of being able to model true beliefs and evaluation methods has an advantage over models which dictate what should be evaluated. This advantage is the opportunity to incorporate irrationalities into a model and an ability to see how that irrationality affects the decision making process.

B. RECOMMENDATIONS

The neural network approach to solving a negotiation problem requires a different method of representing information than other approaches. In a complex problem representation, the network offers to the user relative ease of recognition of the relations inherent within a problem. However, the builder of the network faces a significant challenge. He/she must have an in-depth understanding of all of the elements involved in a problem and their interrelations. Learning paradigms should be explored to help the builder start with a comprehensive negotiation model and let the system learn and adjust itself to new negotiation situations. It is expected that such a learning paradigm would greatly enhance the development speed.

LIST OF REFERENCES

1. Bailey, David and Donna Thompson. "How to Develop Neural Network Applications." AI Expert Magazine, June 1990, 38-47.

2. Bailey, David and Donna Thompson. "How to Develop Neural Network Applications." AI Expert Magazine, September 1990, 34-41.

3. Bui, Tung, and Taracad Sivasankaran, "Fuzzy Preferences in Bilateral Negotiation Support Systems." In Proceedings of the 24th Hawaii International Conference on System Sciences Vol. III, Kola, Hi. 1991, (IEEE, 1991), 687-694.

4. Caudill, Maureen. "Neural Networks Primer." AI Expert Magazine, 1989; reprint, San Francisco: Miller Freeman Publications, 1989.

5. Caudill, Maureen. "Avoiding the Great Back-Propagation Trap." AI Expert Magazine, July 1991, 29-35.

6. Ebehart, Russell and Roy Dobbins, eds., Neural Network PC Tools: A Practical Guide. San Diego: Academic Press, 1990.

7. Fisher, Roger, and William Ury, Getting to Yes: Negotiating Agreement Without Giving In Boston: Houghton Mifflin Co., 1981.

8. Jelassi, M. Tawfik, and Abbas Foroughi, "Negotiation Support Systems: An Overview of Design Issues and Existing Software." Decision Support Systems 5 (1989): 167-181.

9. Jorgensen, Chuck and Chris Matheus. "Catching Knowledge in Neural Nets." AI Expert Magazine, December 1986, 30-38.

10. Kersten, G. E., Wojtek Michalowski, David Cray and Ian Lee, An Analytic Basis for Decision Support in Negotiations. Ottowa: Decision Analysis Laboratory, School of Business, Carleton University, 1989?.

11. Kersten, G. E. and Tomasz Szapiro, "Generalized Approach to Modeling Negotiations." European Journal of Operations Research, 26 (1986): 142-149.

12. McClelland, James, David Rumelhart, and The PDP Research Group, Parallel Distributed Processing, Volume 2, Psychological and Biological Models. Cambridge, Massachusetts: MIT Press, 1986.

13. McClelland, James, and David Rumelhart, Explorations in Parallel Distributed Processing A Handbook of Models, Programs, and Exercises Cambridge, Massachusetts: MIT Press, 1988.

14. Rumelhart, David, James McClelland, and The PDP Research Group, Parallel Distributed Processing, Volume 1, Foundations. Cambridge, Massachusetts: MIT Press, 1986.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexanderis, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Department Chairman, Code AS Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5100	1
4.	Curricular Officer, Code 37 Naval Postgraduate School Monterey, California 93943-5000	1
5.	Professor Tung Bui, Code AS/Bd Naval Postgraduate School Monterey, California 93943-5000	1
6.	Professor Lawrence Williamson Code CSWi Naval Postgraduate School Monterey, California 93943-5000	1